

Small or medium-scale focused research project (STREP)



ICT Call 8

FP7-ICT-2011-8

Cooperative Self-Organizing System for low Carbon Mobility at low Penetration Rates

COLOMBO

COLOMBO: Deliverable 1.3

Decentralized Monitoring based on Low Penetration Traffic Information

Document Information	
Title	Deliverable 1.3 - Decentralized Monitoring based on Low Penetration Traffic Information
Dissemination Level	PU (Public)
Version	1.0
Date	16.11.2015
Status	Delivered

Authors	Jérôme Härri (EURE), Sosina Gashaw (EURE), Luca Foschini (UNIBO), Paolo Bellavista (UNIBO), Wolfgang Niebel (DLR), Andreas Leich (DLR), Marek Junghans (DLR), Robbin Blokpoel (PEEK), Laura Bieker (DLR), Hagen Saul (DLR) Karsten Kozempel (DLR), Thrasyvoulos Spyropoulos (EURE)
---------	--

Table of contents

1	Introduction.....	4
1.1	Document Objectives	5
1.2	Document Structure.....	5
2	Decentralized Traffic State Estimation.....	6
2.1	Cluster-based Fusion for Traffic State Estimation	6
2.2	Dissemination-based Fusion for Traffic State Estimation	7
2.3	Fusion-based Traffic State Estimation (TSE)	19
2.3.1	Method	19
2.3.2	Experimental Results	28
2.3.3	Discussions & Future Prospects.....	37
3	Further Decentralized Surveillance Applications.....	38
3.1	Driving Anomalies and Hazard Detection	38
3.1.1	Identifying Atypical Behaviour: Review of the State of the Art	39
3.1.2	The Probability Density Map (PDMaP) Approach	40
3.1.3	Experimental Setup and Investigational Approach.....	43
3.1.4	Experimental Results	48
3.1.5	Discussion and Future Prospects.....	51
3.2	Local Emissions Monitoring	51
3.2.1	Simulation Scenario	52
3.2.2	Emission Monitoring Approach.....	53
3.2.3	Simulation Results	62
3.2.4	Discussions.....	63
4	Conclusion	64
	Appendix A – References	67
	Appendix B: Parameterization of Kalman Filter	71

1 Introduction

The COLOMBO project will deliver a set of modern cooperative traffic surveillance and control applications that target at different transport related objectives, such as increasing mobility, resource efficiency, and environmental friendliness.

COLOMBO develops methods for tactical traffic surveillance, which allow gathering information about the current state on the roads at low costs. This is mainly achieved by exploiting data that is available from vehicular communication (V2X) that is aimed to be introduced in 2015.

COLOMBO therefore develops traffic surveillance systems falling in a new category called '*Distributed Floating Car Data*' (DFCD). In DFCD, vehicles and smartphones are sampling traffic data, but instead of transferring these data to a central processing server, they use V2X technology to autonomously compute local traffic information and transmit them directly to traffic lights. The main objective of the local computation is not to reduce traffic on the communication and cloud systems, but to be able to work without it.

The unique aspect of COLOMBO is to consider early deployment of V2X technologies, where the number of V2X equipped vehicle will not have reached a critical mass for typical DFCD as those proposed in [COLOMBO D1.2, 2014]. The limits of these DFCD algorithms have been emphasized and described, notably their instability when V2X penetration goes below 10%.

Accordingly, the objective of this document is to move one step further and propose enhancements to DFCD by using alternate technologies or devices. Limitations in DFCD may come from two aspects:

Limited and heterogeneous knowledge of GNSS positions – most of the DFCD algorithms described in [COLOMBO D1.2, 2014] are based on vehicles knowing their locations and that of their neighbours. GNSS technologies are however only capable of providing between 2-7 m precision in the best conditions. Accordingly, GNSS-based traffic state information are expected to be approximate. Also, relaying on alternate devices (smartphones) to compensate for low penetration V2X technologies will add further uncertainties to GNSS information. The main reason is that GNSS technologies are not expected to have the same level of precision when equipped on-board of vehicles or in smartphones. This will lead to a limited and heterogeneous knowledge of GNSS-based information that COLOMBO DFCD will have to correct.

Reduced Sensing (detection) capabilities – Although V2X technologies are expected to require a decade to become widely spread in vehicles, alternate communication technologies are already available, such as Bluetooth (BT) or WiFi-Direct. They however do not provide the same communication and traffic sensing capabilities as V2X technologies. WiFi-direct notably lacks a proper ad-hoc multi-hop communication mode, while Bluetooth has a significantly smaller communication range, as well as limited sensing capabilities as cruising speed. This will require DFCD extensions to support reduced data or heterogeneous communication ranges.

The common aspect behind these two DFCD limitations is the need avoid single source of information, but instead to rely on a robust fusion engine capable of filtering outliers, predicting missing data samples, and integrating data of heterogeneous quality. This will be the first task of this document. The second task of this document will be to use COLOMBO resilient DFCD to provide rich Smart Mobility applications, such as:

Anomalies and Incident Detection (AID) – although urban traffic is traditionally well understood and controlled, deviation from the general traffic flows may be observed, either from minor traffic anomalies to more serious traffic incidents. COLOMBO DCFD may also be used to detect such situation, and advanced data inference-based models will be proposed to efficiently detect traffic anomalies and incident, even at low V2X penetration.

Local Emission Modelling – urban pollution is growing toward being a major source of health issues, as more population live and work in urban centers. Accordingly traffic infrastructures must help vehicles (drivers) adopt a pollution-saving attitude. A major innovation may come from dynamic adaptations of traffic lights not only based on traffic jams but on pollution levels. Basic emission sensors are not capable of extracting a representative image of the vehicular emission. Accordingly, COLOMBO will propose a DFCD-based approach capable of using traffic state information, even at low V2X penetration rates, to provide a precise estimation of the local emissions around urban intersections.

1.1 Document Objectives

The objectives of this deliverable are twofold: built from [COLOMBO D1.2, 2014], alternate technologies will be proposed to compensate low penetration of Class C (i.e. V2X). Second, illustrate the benefit of local traffic state estimations for driving anomalies detections and local emission monitoring. This document builds the bridge with Swarm-based smart traffic lights work conducted in WP2.

More specifically, this document will describe advanced DFCD supporting low penetration of V2X technologies:

- **WLAN-V2X Fusion-based DFCD** – the document will describe how fusion of low penetration Class C (V2X) data with a larger penetration of Class B (WiFi-direct) data (considering the drivers' smartphones for instance) will manage to compensate the reduced DFCD quality observed on [COLOMBO D1.2, 2014] for low penetration of Class C (V2X) vehicles.
- **Bluetooth-V2X Fusion-based DFCD** – the document will describe how fusion of low penetration of Class C (V2X) data with a larger penetration of Class B (BT) data (considering vehicles on-board Bluetooth (BT) technology) will also manage to compensate the reduced DFCD quality observed on [COLOMBO D1.2, 2014] for low penetration of Class C (V2X) vehicles.
- **Traffic Anomalies Detection**– the document will describe how DFCD, even at low penetration rate, will be able to provide sufficient traffic information to detect hazardous traffic situations. This work has been based on real traffic monitoring in order to observe real traffic anomalies.
- **Local Emission Monitoring** – the document will finally describe how DFCD, even at low penetration rate, is capable of providing emission information to a RSU, which is very close to realistic measured conditions. This work is therefore capable of reducing the need to have manual measurements, and adds flexibility to getting emission data in locations or in contexts that cannot be otherwise measured.

1.2 Document Structure

In Chapter 2, fusion-based decentralized low penetration traffic state estimation systems (DFCD) are described, ranging from Class B (WLAN) to Class B (BT) assisting low penetration Class C (V2X) vehicles. Afterwards, Chapter 3 introduces and evaluates further decentralized surveillance applications, such as traffic Anomalies and Incident Detection (AID) as well as Local Emission Monitoring (LEM) systems. Chapter 4 summarizes the described work.

2 Decentralized Traffic State Estimation

2.1 Cluster-based Fusion for Traffic State Estimation

The vehicles that participate in the traffic monitoring algorithm can have a complete communication package (Class C) or only a limited capability provided by an application running on a PDA (Class B). These different vehicle types differ not only in their communication capabilities, but also in the precision on the sensors used to collect the positional information. As the precision of the Cluster-based Traffic State Estimation described in [COLOMBO D1.2, 2014] significantly depend on the vehicles to precisely know their locations, heterogeneous position information (various GPS quality between V2X and Smartphones) is expected to play a critical role.

To model these differences, we use a configurable class that, starting from the real positional data from sumo, introduces random errors of different magnitude according to the type of vehicle modelled. This class can tamper with the vehicle position, its speed and its direction.

The information obtained from these different types of vehicles is then shared with the RSUs by the communication protocol and fused together to create an overall estimation of the traffic status. While on the one hand the use of lower quality information collected from vehicles with limited capability lowers the accuracy of the overall estimation, on the other hand it provides a larger dataset that can be used to draw the estimation, improving its quality compared with one obtained using only data from full capability vehicles.

The two chart below depict the number and the average speed of the vehicles in the scenario Andrea Costa for different mix of vehicles types. With the difference of the gradual positioning error, the simulation parameters are similar to that described in [COLOMBO D1.2, 2014].

While the number of vehicles show significant difference between them, mainly due to the lower range of the degraded vehicles, the average speed does not show significant differences.

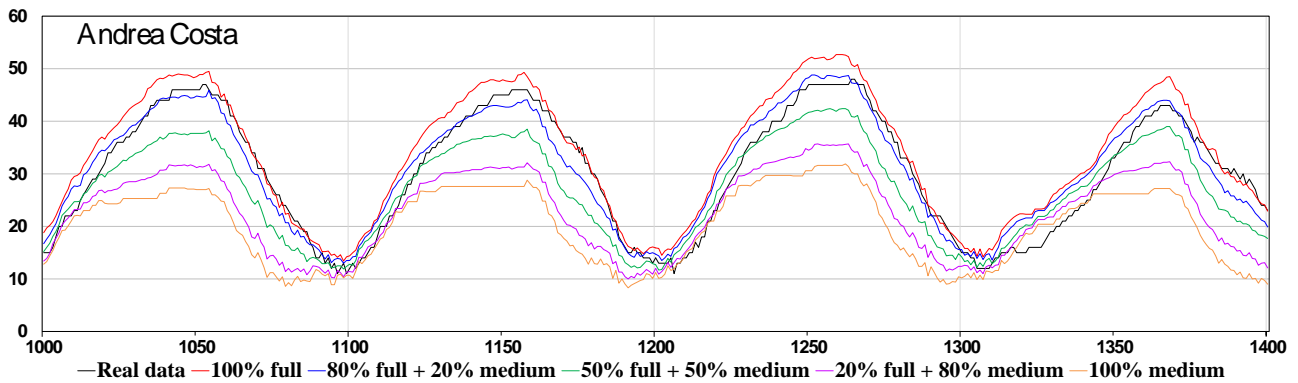


Figure 1: Number of cars in Andrea Costa with different mix of vehicles types

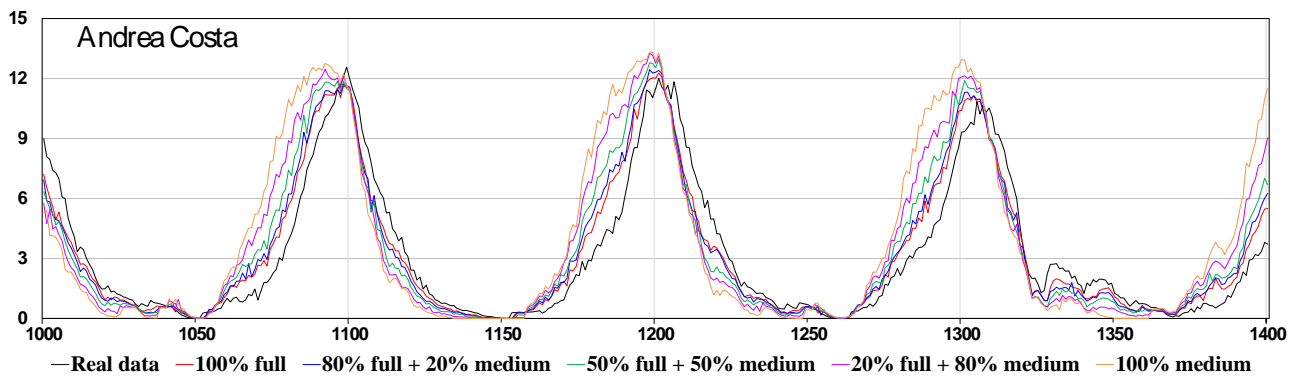
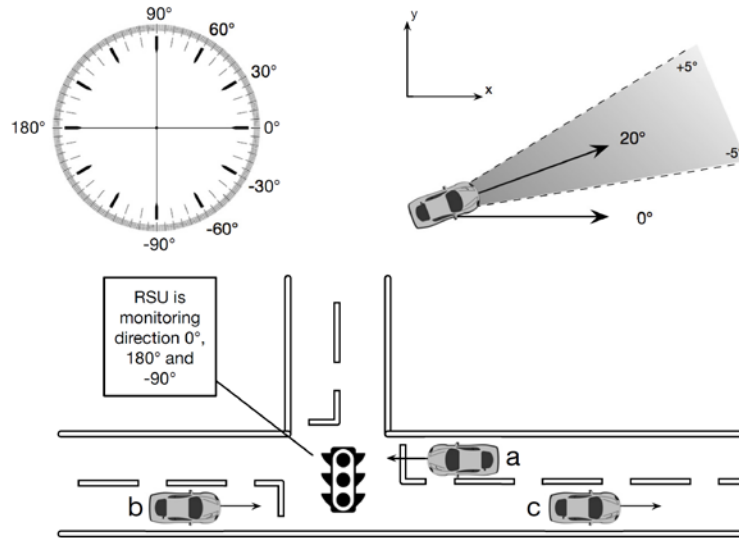


Figure 2: Average speed in Andrea Costa with different mix of vehicles types

Then, to locate a vehicle near an intersection, when GPS shows errors, map matching may be employed. We can use the positional information in different ways: we can use the vehicle direction to estimate which road it is currently travelling on; we can use its position and correlate it with a map to find the edge that it is using.

In the first method for every monitored edge incoming or outgoing an intersection, the RSU assigns it a direction. To find it is travelling on an edge a vehicle compares its direction with the one assigned to the edge to check if it is inside a tolerance error angle. Since on a road the edge leading to the intersection has the same direction as the one leaving it, the direction information used to identify an edge also has a field to discriminate between them.



The second method assumes that the vehicles have a map of the road network, and know the topology of the streets that they will travel. To check if it is traveling on a particular edge, a vehicle computes the distance from its current position and the vertex of each edge in its neighbourhood. The vehicle then selects as its edge the one with the lowest distance computed before.

There is also a third method, which is the union of the above-mentioned two. A vehicle first estimates its direction using a larger error angle compared to the one used in the first method, and then it uses the second method to estimate the edge on which it is travelling on. The overall information is then the union of the two estimations.

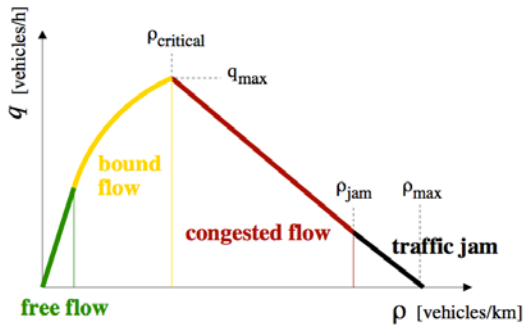
Detailed performance evaluation of the three methods are provided on [COLOMBO D5.4, 2015].

2.2 Dissemination-based Fusion for Traffic State Estimation

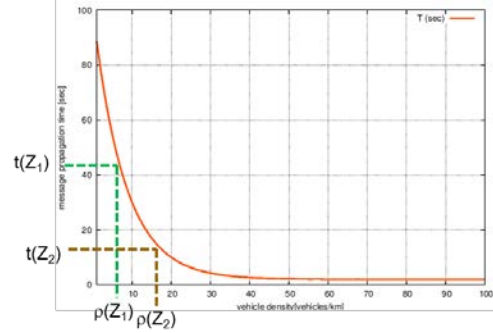
The traffic state estimation algorithm described in this section is an extension of the DissFlow protocol described in [COLOMBO D1.2, 2014], extended and adapted to support Class B and Class C vehicles, as well as less homogeneous traffic condition. We briefly remind here the basic features on which DissFlow is built, as well as specific extensions for heterogeneous technology and traffic flows, and let interested readers to [COLOMBO D1.2, 2014] for more details.

Similarly to the fact that Traffic Flow theory is based on three fundamental diagrams, one of which being illustrated in Figure 1(i), data dissemination is also based on one major fundamental diagram. Depicted on Figure 1(ii), it shows the relationship between the delay in multi-hop dissemination wireless networks and the network density. The denser it is, the lower is the delay. DissFlow has

been designed on the proposal to revert this relationship and compute network (urban road) density from multi-hop dissemination delay.



(i) Traffic Flow



(ii) Data Dissemination

Figure 3: Fundamental Diagrams

As it can be observed on Figure 4, DissFlow has two phases :

- **Traffic Monitoring Notification** – The RSU notifies all vehicles entering a particular part a road network to start the DissFlow monitoring. All vehicles already in or still out of the designated area are not influenced.
- **Traffic Monitoring Report** – Upon the initiation of the traffic monitoring phase, one or more traffic monitoring messages are sent by vehicles back to the RSU. These messages are carried in the case of low density or relayed in case of high density, and once back at the RSU, it reports various traffic related data: *dissemination delay*, *mean number of neighbors* and *length of longest connected cluster*.

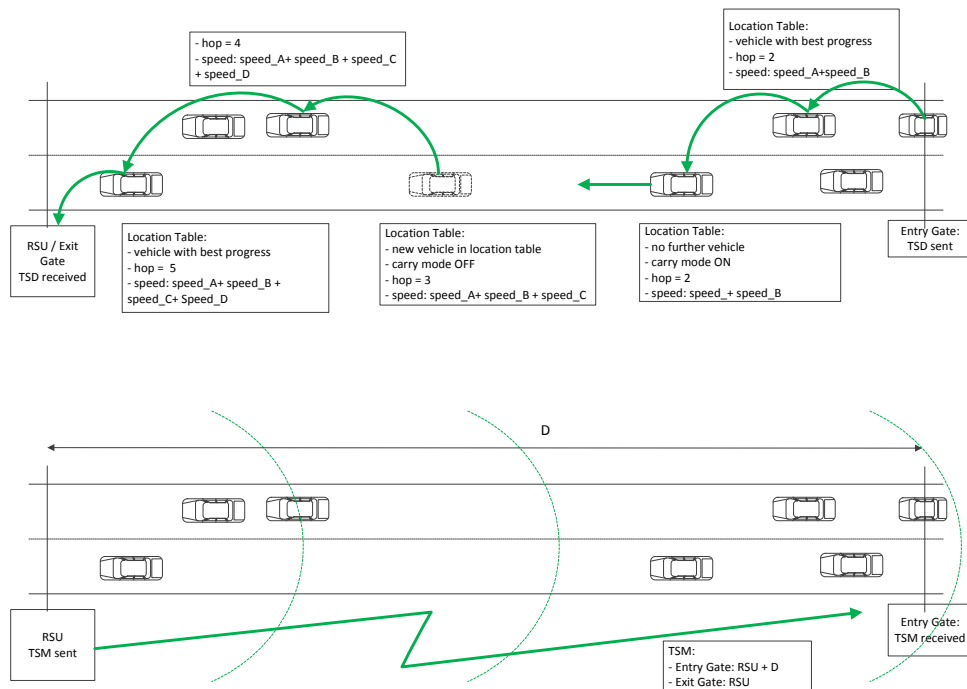


Figure 4 DissFlow Protocol Description

A key aspect of DissFlow is therefore its dissemination algorithm, which should be able to support various traffic density, and also different technologies. It should be noted that unlike the definition of Class C and class B vehicles differing by their sensing capabilities and GPS positioning, our definition of the Class B and Class C vehicles defer only by the communication technology available and not on their capabilities to know their position.

In this work, we assumed Class B vehicles to be equipped with Bluetooth technology. The main reason behind this choice is the larger penetration of Bluetooth on board of vehicles as well as its robust communication capabilities. WiFi-Direct would have been another valuable option, but two limitations have been experimentally observed:

- WiFi-Direct does not support multi-hop ad-hoc communication, due to the strict control and separation of WiFi groups by WiFi Group Organizers.
- WiFi-Direct suffers from a high delay in establishing an ad-hoc link (p.ex 2-3 s), which would significantly impact DissFlow.

In the rest of this section and unless otherwise indicated, Class B vehicles will be considered as equipped with Bluetooth technologies.

We will first remind the multi-hop relaying protocol behind DissFlow and its implementation on iTETRIS, then we will describe the three traffic monitoring modes of DissFlow, and finally provide performance evaluations, first in terms of absolute traffic states metrics, and also in terms of Root-Mean-Square Error metrics (RMSE).

2.2.1.1 DissFlow Dissemination Algorithm

DissFlow dissemination algorithm operates as depicted on the Flow-chart on Figure 6. Upon reception of a Traffic State Data (TSD) message, a vehicle will look for relaying candidates in its Neighbor, reps. Location Table. If none is found, the TSD message is carried until either one is found or the RSU is reached. Otherwise, the vehicle selects the vehicle providing the maximum progress on the same driving direction.

It should be noted that this DissFlow Flow-Chart applies regardless of the technology used (Class B (BT) or Class C (V2X)), as both technologies have the table concept listing neighbours in their range. One assumption yet is that even the Bluetooth technology is capable of transmitting the GPS (or related) position. We expect this to be possible in future release of automotive-grade Bluetooth.

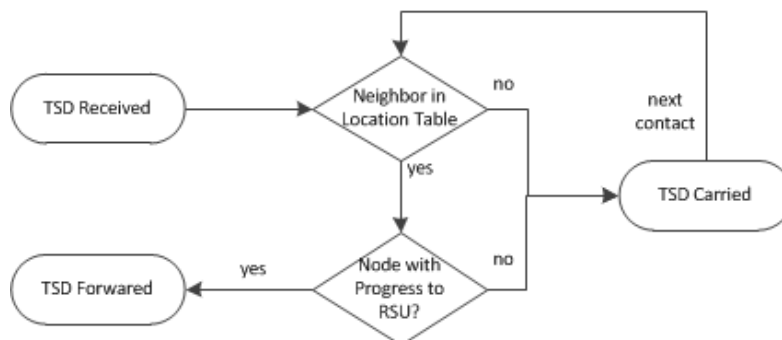


Figure 5 DissFlow Flow-Chart

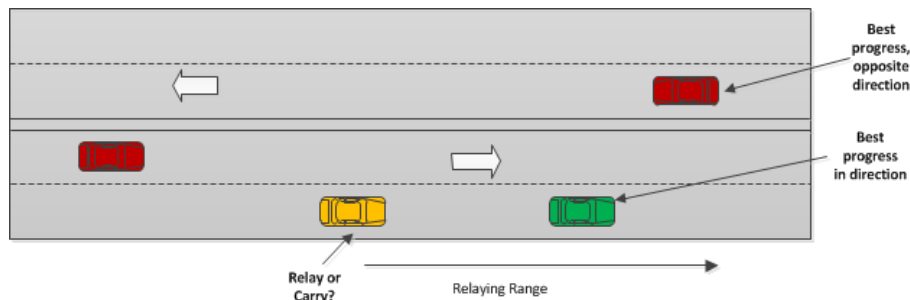


Figure 6 DissFlow Relaying Options

DissFlow has been implemented on the ns-3 side of iTETRIS, by adding the following features to the iTETRIS ETSI ITS stack (see Figure 7):

- A Carry-mode – if a neighbour is not found, the packet is not dropped but rather carried.
- Extended Geonet header – The ETSI GeoNet header already provides support to exchange GPS position and speed. Yet, DissFlow requires three additional data fields:
 - **Transmit time** – Time Stamp when the original packet has been sent, regardless of the number of relays.
 - **Mean Speed** – moving average of the velocity observed by all relaying vehicle.
 - **Mean Neighbourhood size** – moving average of the size of the neighbourhood by all relaying vehicle
 - **Connected Cluster** – the number of hop since the last carry mode. It should be noted that it differs from a basic hop count, which would instead provide the total number of hops.

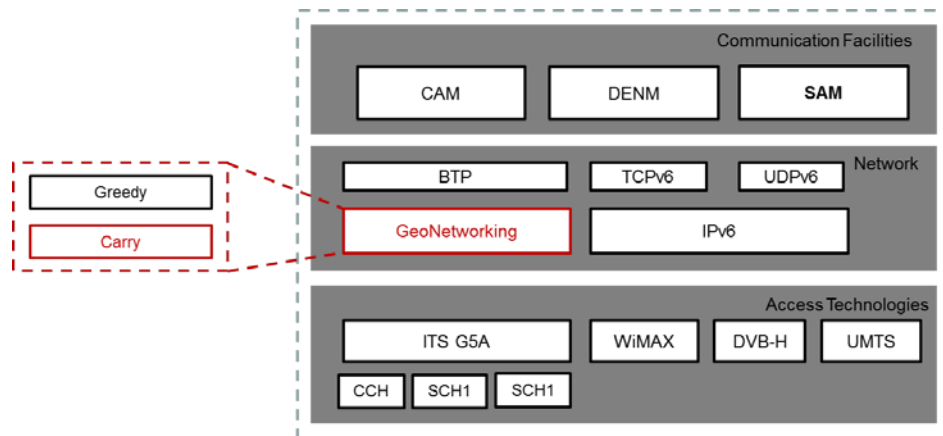


Figure 7 DissFlow iTETRIS integration

2.2.1.2 DissFlow Traffic State Estimation Modes

On D1.2, DissFlow is described as having the dissemination delay as being the single traffic state estimation mode. Yet, TSD messages carry additional information while being relayed back to the RSU, which can also be used. Having limited impact for homogeneous traffic environments, the additional traffic monitoring modes become critical when traffic becomes less regular.

DissFlow has the following traffic monitoring modes:

- **Delay** – as described in D1.2, DissFlow extracts traffic density from the delay required by packets to be disseminated back to the RSU

- **Neighbour** – this mode reports the mean number of neighbours of each relaying/carrying vehicle. This is a representation of the traffic volume observed by relaying vehicles and therefore can be used to estimate traffic state.
- **Cluster** – this mode reports the mean length (in number of vehicles) of the connected cluster behind a vehicle reaching the traffic light. It is obtained by letting the first vehicle starting resuming a relay (after a carry mode) to count the number of relay hops before reaching the RSU. If a packet is carried again before reaching the RSU, the # hops is kept in a moving average. The length of the connected cluster reaching a traffic light is a critical metric representing the number of vehicles approaching it.
- **Speed** – this mode does not report traffic volumes but rather traffic speed estimates. It measures the mean speed of each relaying vehicle and is in case of multiple hops or carrying phases, it is capable of reporting multiple samples of observed vehicular speed.

TSD messages sent by DissFlow and reaching a RSU carry all necessary data to support the four modes, although their individual performance will strongly depend on the number of hops, the carrying duration and the transmit range.

2.2.1.3 Simulation results

We evaluate the performance of DissFlow in two different scenarios. The first one corresponds to a 1D urban road, which does not include traffic lights. Accordingly, the inter-distance between vehicles may be controlled and kept close to the theoretical estimation. The second scenario corresponds to the COLOMBO RiLSA scenario, which consists of a single intersection with traffic lights. This scenario is more ambitious for DissFlow, as the traffic lights will significantly alter the inter-distance distribution between vehicles.

2.2.1.3.1 1-D urban road

The scenario corresponds to a 1km long straight urban road, including between 1 to 3 lanes as illustrated in Figure 8. The traffic arrival may be controlled by an Erlang distribution in order to keep exponential inter-arrival time between vehicles. The scenario actually 3km, but we remove the first and last km in order to mitigate border effects. The simulation parameters are indicated in Table 2.1

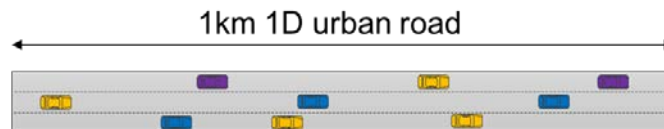


Figure 8: 1-D Urban road scenario

Table 2.1 DissFlow Simulation Parameters

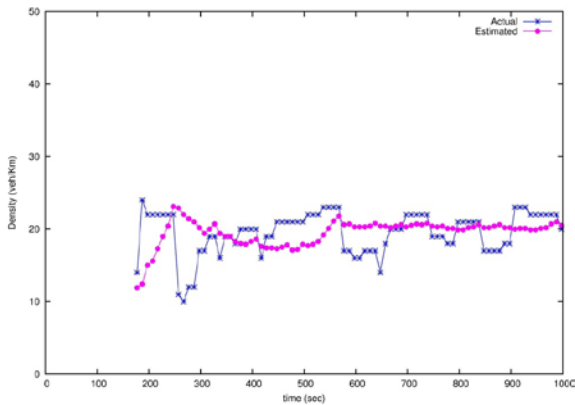
Type	Value
technology	ITS-G5 / IEEE80211.p
frequency band	10MHz @ 5.9GHz
Fading	log-distance, $n = 2$
Tx power	12 dBm / Tx range = ~150m
TSD/TSM size	100 Bytes
scenario length	1km 1-D urban road

mean inter-arrival	$\mu = 2s$
speed (S_d)	constant, lane specific [20m/s - 40m/s]
Relay Speed (S_c)	$3 \cdot 10^8$ m/s
filter length (l)	100 samples
simulation time	1600 s

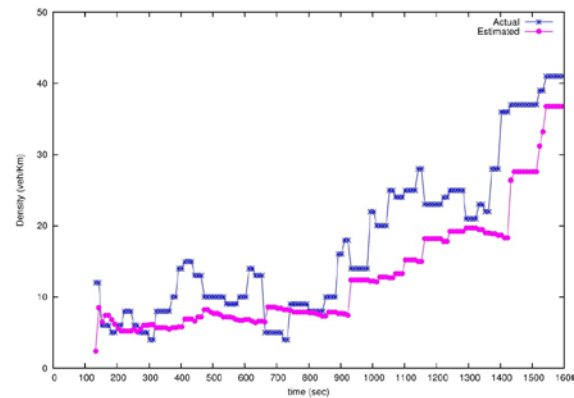
We evaluate here the reliability of DissFlow to compute traffic density estimates. In order to model more realistic urban vehicular mobility, traces are extracted from SUMO injected into iTETRIS/ns-3. The vehicular arrival rate are yet still configured to follow a Poisson distribution, although its variance is significantly reduced in the high density case. We compare the traffic density estimates against the ground truth provided by SUMO.

In a first step, we keep the traffic density stable (i.e. constant μ) in order to test DissFlow ability to reflect traffic. In a second phase, we make traffic density vary in order to evaluate the DissFlow capability to follow the traffic trend. In order to reduce the variability of the multiple density estimates received by the RSU, we filter the delay samples with a moving average over 100 samples.

The first set of results depicted in Fig. 8 relates to the first strategy. Fig. 8a depicts the raw samples received by the RSU, while Fig. 8a shows the filtered traffic density estimates. We can observe that DissFlow reproduces the traffic density, although at some occasions, it does not react to small varying states. Having a look at the raw samples in Fig. 8a, we can see that this comes from the filtering phase as well as from sparse samples. In Fig. 9, we address the second strategy. Once again, we can observe that DissFlow does not react to small oscillation of traffic density. DissFlow is yet capable of following and matching the gradually and long term increase of the traffic density. From the raw samples, we can also observe an increasing variability of the delay samples, due to an increasing role of the relaying phase over the carrying phase. We can yet also observe that DissFlow underestimates traffic density.



Class C stable density



Class C varying density

Figure 9 Traffic Density Estimation - Class C

Overall, DissFlow shows to be able to fairly estimate traffic density. It does not react for brief traffic density variations, but follows the trend when the traffic shows an steady increase. It yet tends to underestimate traffic and show variability at increased traffic density. These could be explained by the hypothesis of the delay/density model, which assumes Poisson distributed inter-

vehicular distances, and which are not found at high traffic density. Although the smoothing factor of DissFlow could be seen as beneficial to TLC systems, the filtering model will be subject to future studies to react closer to traffic density fluctuations.

The overhead of DissFlow is measured by the number of required TSD messages, and corresponds to $\approx 30\text{bytes/second/vehicle}$ at the highest density, which corresponds to 1/50 of the CAM overhead.

2.2.1.3.2 Rilsa

The second scenario is called COLOMBO Rilsa [47] and is illustrated on Figure 10. Traffic is configured to realistically reflect urban intersection traffic, and has been used as a baseline traffic scenario in COLOMBO.

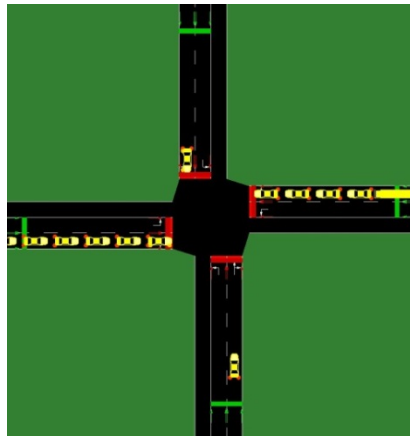


Figure 10: Simulated intersection with 4 arms (COLOMBO RiLSA 1 example [47]).

Beside the different scenario, Class B as well as Class C vehicles will be evaluated. The objective will therefore be to evaluate the capabilities of moderate penetration of Class B vehicles to counterbalance the low penetration of Class C vehicles. In this work, Class B vehicles are assumed to have on-board Bluetooth capable of data relaying up to 15m. Bluetooth has not been simulated on ns-3, but emulated by a shorter range ITS-G5. Other parameters are described on Table 2.2.

Table 2.2 RiLSA – Dissflow Simulation Parameters

Type	Value
technology	ITS-G5 / Bluetooth
frequency band	10MHz @ 5.9GHz
Fading	log-distance, $n = 2$
Tx range	100 ITS-G5 / 15m Bluetooth
TSD/TSM size	100 Bytes
Traffic scenario	RiLSA
filter length (l)	100 samples
simulation time	1600 s

We finally evaluated not only Dissflow capabilities to estimate traffic density but also traffic speed. Considering that RiSLA does not reproduce exponential inter-distance between vehicles, DissFlow density-based mode needs to be complemented by two other modes: cluster and neighbour.

Traffic Volume Estimations

In the first set of results, we evaluated DissFlow's capability to evaluate the traffic volumes considering class C vehicles first, then class B vehicles equipped with Bluetooth technologies, and finally jointly class C and class B. We also evaluate each of Dissflow mode, although the Delay mode is rarely efficient in the RiSLA scenario. We evaluated the penetration ratio in the range of [10,20,50,100%].

Figure 11 depicts the separate evaluation of the three Dissflow modes, where the red line represents the ground truth obtained by SUMO. We can observe that all modes closely follow the ground truth at 100% penetration, but then experience a general gap. Although traffic density fluctuations created by RiLSA are followed by both Neighbor and Delay modes, they significantly overestimate the true density. We do not classify this as major drawback, as once the drift known, fluctuations are still followed by DissFlow. At penetration rate below 20%, the traffic volumes stop being accurate. This is a clear example of the limits of low penetration of V2X technology for traffic surveillance.

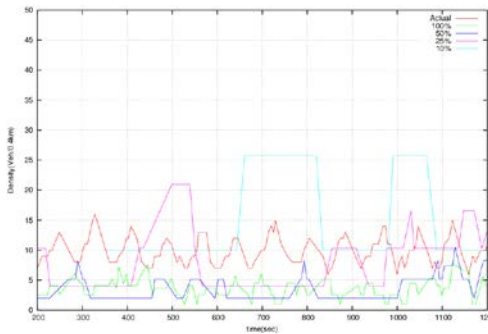
We could also mention that the tendency to overestimate traffic density has also been observed by other approaches investigated in COLOMBO, which comes from the general tendency of counting multiple times the same vehicles in the measurement area.

One way of mitigating this limitation is to statically or dynamically define zones, where traffic states are measured once and only once. This is a direction for future work.

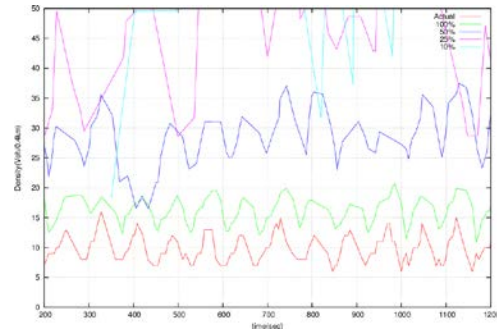
Then Figure 12 illustrates the fusion of traffic state estimation from the three DissFlow mode. As previously described the fusion is based on a learning phase, where the errors are nullified by proposer weighted filters. Then the weighted filter is used on unknown traffic.

As it can be observed, we generally manage to reduce the error, having 100% penetration closely following the traffic density in terms of trends and absolute values, 50% and 20% following the trends only, while lower penetration are not accurate enough.

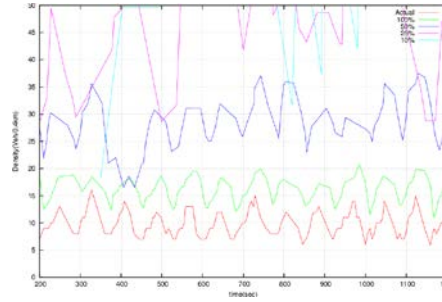
We provide in Table 2.3 the RMSE evaluation of each of the DissFlow modes, including the learning-based fusion approach. We can clearly see from the RMSE that fusing various modes manages to reduce the RMSE at at least a factor 2. This comes from the beneficial influence of the cluster mode, which has a low RMSE and as such manages to compensate the others.



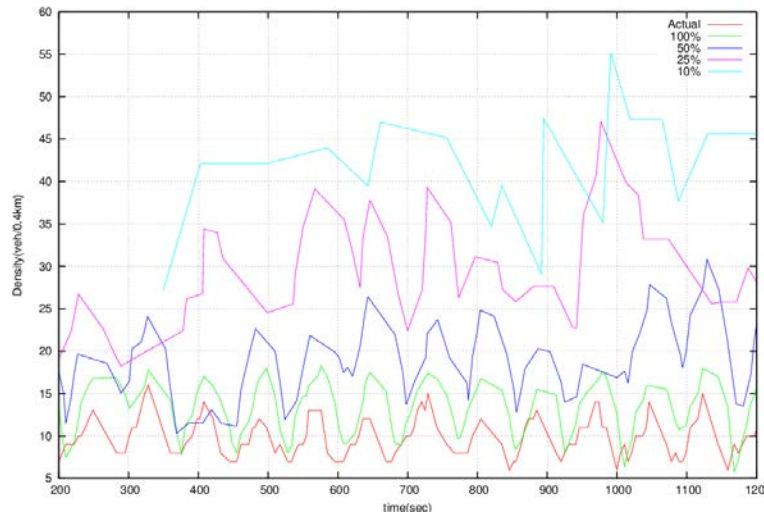
(i) Cluster mode



(ii) Neighbor mode



(iii) Delay mode

Figure 11 Class C – Impact of the three DissFlow modes, at various penetration rates**Figure 12 Class C - Fusion of DissFlow modes****Table 2.3 Traffic volume RMSE for Class C individually, considering various penetration rates and DissFlow modes.**

Class C (V2X)	Penetration	Delay	cluster	Neighbor	Fusion
	100%	6.41	6.84	11.35	3.91
	50%	18.98	6.45	16.03	9.78
	25%	39.06	6.15	22.28	19.63
	10%	61.47	9.00	43.36	35.79

We then illustrate in Figure 13 the traffic density estimation considering Class B vehicles, say Bluetooth technology (BT), for varying penetration rates [10%,20%,50%,100%]. We can also see that accuracy of traffic volume estimations also degrades at lower penetration.

Although BT benefit from a larger presence on board of vehicles, its low transmit range remains a critical drawback to DissFlow. Although trends behind traffic volumes may be followed for 100% and 50% penetration, traffic estimations at lower penetration rate strongly diverge.

Fusing the different DissFlow model does not help as much as expected, as in this case, the learning phase can never find sufficiently stable patterns to be reproduced after the learning phase.

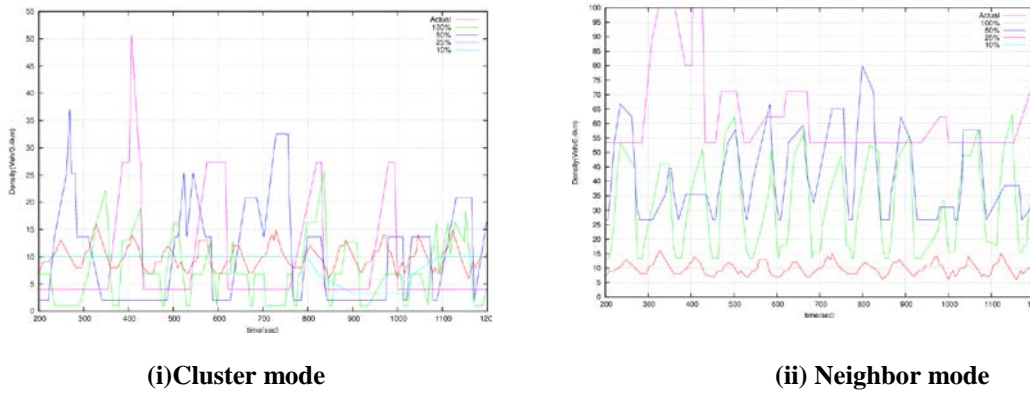


Figure 13 Class B - Impact of the three DissFlow modes, at various penetration rates

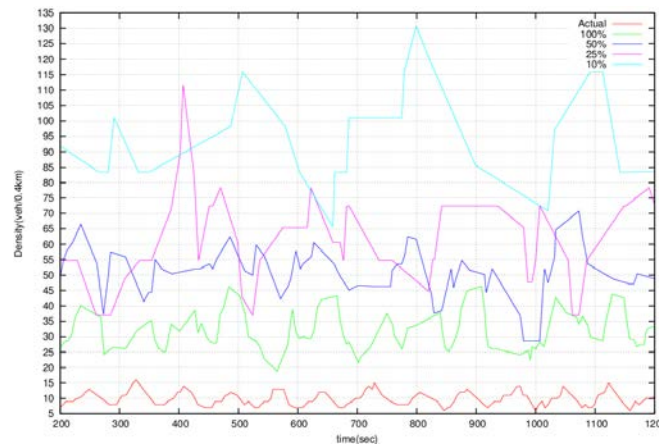


Figure 14 Class B (BT) - Fusion of DissFlow modes

We provide in Table 2.4 a summary of the RMSE of the different DissFlow modes for traffic density estimation. As illustrated, two observation may be done: first, only the DissFlow cluster-mode should be used for Class B (BT), and second a rather high penetration of BT should be assumed in order to be able to counter-balance the low penetration of Class C (V2X) vehicles.

Table 2.4 RMSE of traffic volume for Class B vehicles individually, considering various penetration rates and DissFlow modes

Class B (BT)		Delay	cluster	Neighbor
	100%	52.4	5.93	42.5
	50%	96.0	10.66	34.21
	25%	11.0	16.7	62.84
	10%	117.36	26.45	144.95

In Table 2.5 we fusion traffic density estimates of Class C (V2X) and Class B (BT). The good performance of the DissFlow cluster mode is confirmed, as well as the benefit of relying on a large penetration of Class B (BT) vehicles (above 50%) to assist low penetration of Class C (V2X) vehicles.

Table 2.5 RMSE of traffic volume - fusion between Class B and Class C vehicles

Class C (V2X)	Delay	cluster	Neighbor
50%-50%	41.73	6.15	33.96
25%-25%	84.10	12.7	47.6
10%-10%	99.01	21.3	115.8

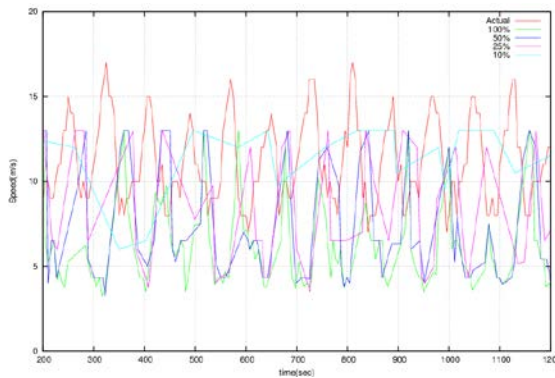
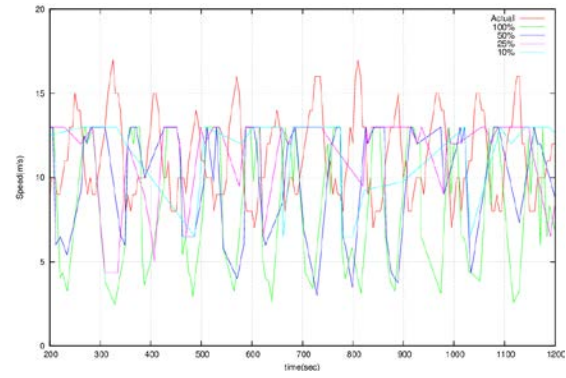
Traffic Speed Estimations

In this part, we now evaluate DissFlow capabilities to estimate traffic speed rather than volumes. As we could observe in the previous results, as well as confirmed by other TSE approaches in COLOMBO, traffic density is more difficult to estimate than speed. Yet, the COLOMBO Swarm traffic light control systems are fully capable of adapting traffic light cycles based on traffic speed rather than density.

Accordingly, we evaluate in Figure 15 DissFlow's performance to estimate traffic speed, first considering Class C (V2X) vehicles only, then Class B (BT) vehicles only, and finally proposing to fusion both estimations at varying penetration rates.

As expected, results closer follow the ground truth estimation. Unlike density estimation, traffic speed is not over-estimated., but the challenge is for the monitoring algorithm to follow quick fluctuations of high and low speed controlled by traffic lights.

We can observe that up to 25% penetration, the fluctuation of traffic speed may be fairly estimated and for both Class B and Class C vehicles. Although curves do not seem to be closely matching the ground truth, we can actually observe a drift in the speed estimates. This means that the curves matches values and trends but with a small delay due to the dissemination-nature of DissFlow.

**(i) Class C vehicles****(ii) Class B vehicles****Figure 15 Traffic Speed Monitoring – Multi-Class, at various penetration rates**

The good performance of DissFlow for traffic speed estimations is confirmed by the RMSE depicted in Table 2.6, although low RMSE for low penetration should not be misled by good performance. Nevertheless, this allows us to another observation, also seen in other TSE described in COLOMBO: the RMSE is decreasing while penetration increases.

Although more details investigations would be necessary, this unexpected feature may be explained by the flattening of the speed curves at lower penetration. Indeed, at high penetration, DissFlow

follows the trends (ups and downs) of the speed fluctuations with minor errors when normalized. But at lower penetration, Dissflow tends to provide flat (constant) speed estimates, which then tends to reduce the RMSE although it does not fully represent accurate speed estimates.

Table 2.6 Traffic speed for Class B and Class C individually, considering various penetration rates

	100%	50%	25%	10%
Class C (V2X)	5.13	4.65	3.65	2.98
Class B (Bluetooth)	3.35	2.70	2.18	2.31

In the final table of this evaluation, we report the fusion of traffic speed estimates between Class C (V2X) and class B(BT) vehicles. As for density estimates, we can also clearly observe the beneficial factor of using alternate technologies for traffic speed estimates available at a higher penetration rate in order to compensate for low penetration of V2X technology.

This benefit is not only visible in Table 2.7, where a typical fusion between Class C (V2X) available at 10%, and Class B (BT) available at 30% provides a reduction of the RMSE of both technologies considered individually, as well as on Figure 16.

Table 2.7 Traffic speed - fusion between Class B and Class C vehicles

Class C & Class B	50%-50%	25%-25%	10%-10%	10% - 30%
	3.351	3.35	3.37	2.48

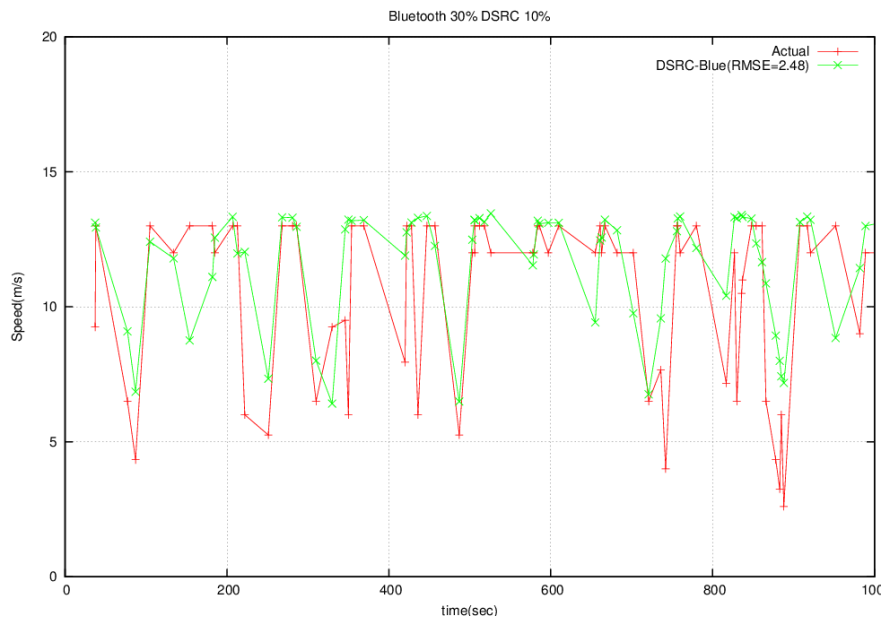


Figure 16 Traffic Speed - Fusion between class B & class C vehicles

2.3 Fusion-based Traffic State Estimation (TSE)

Traffic engineers and managers are interested in high quality traffic data. For instance, in order to have a well performing traffic light control, vehicle speeds and counts are of particular interest. These parameters are provided by induction loop detectors, camera sensors and other technologies. The drawback of current sensors is that they require high installation and maintenance costs (e.g. loop detectors) or they do not provide accurate data or are weather or illumination dependent (e.g. cameras). Therefore, there is a need for other sensors or technologies, Car2X being one of them.

It is expected to take years to achieve an acceptable penetration grade of Car2X [cf. COLOMBO D6.3]. It is a challenge to obtain high quality velocity (and positional data) and reliable counts of the vehicles approaching an intersection, particularly taking into consideration low penetration rates of 1% to 3%. However, answers to the question “How good C2X can be used for traffic detection and for traffic management purposes instead of other established sensor technologies?” can be given, since methods and ideas exist to cope with the sparsity of C2X data. In order to improve the detection accuracy of the incomplete data and to increase the detection horizon it is reasonable to involve other data sources available at or near the intersection. As mentioned above, typically, induction loops and sometimes cameras are available at intersections and can be used for this purpose. However, in this study a different path is taken, paying attention to other wireless communication technologies to find ways to obtain accurate and reliable estimations of vehicle counts and velocities. In particular we concentrated on Bluetooth technology, which has been widely used by traffic participants and is available in Smartphones, hands-free communication systems, navigation systems, radios, laptops and other devices, integrated in vehicles, carried by pedestrians, bicyclists, etc. Depending on the road type and type of the traffic participant, the Bluetooth penetration rate differs from 3 to 50% depending on the application context, i.e. urban or suburban areas, motorways, the amount of trucks, etc. [49], [50], [51] resulting in higher detection ratios in comparison to the sparse C2X data.

This section reports on the development and investigation of a method for velocity and vehicle count determination in case of low penetration C2X and moderate penetration Bluetooth rates for Traffic state estimation (TSE). Therefore, this section contains an explanation of the fusion of rare C2X data with more frequent Bluetooth data (section 2.3.1) and experiment based simulation results (section 2.3.2) of a 4-arm intersection of the COLOMBO simulation scenarios are given. Finally, conclusions and prospects of our future work are given (see section 0).

2.3.1 Method

C2X is a wireless communication technology standard, which is made for traffic and transport surveillance and may be even used for traffic management purposes. In contrast, Bluetooth was not made for traffic management, but has been widely used to traffic detection for years [39], [49]. Within the scope of this study we try to analyse the fusion of Bluetooth with C2X. Due to the sparsity of C2X data not only powerful methods to cope with this lack of data are needed, but also additional sensors are necessary to fill data gaps. For that reason, we developed and tested a method capable of handling incomplete sparse data with great uncertainty to obtain reliable and accurate speed information and vehicle counts.

The approach requires the implementation of the following sub-processes, which are explained in detail in the following:

- Speed and vehicle number estimation
- Sensor data fusion of C2X and Bluetooth data
- Learning
- Transport mode detection

- Direction filtering

2.3.1.1 Speed and Vehicle Count Estimation

In case of C2X among other data and information speed information is broadcasted out and received by the C2X-RSU using the implemented CAM (Cooperative Awareness message) protocols of C2X. The number of vehicles equipped with C2X can hardly be estimated due to the sparsity of the data. In case of Bluetooth, the velocity of a moved Bluetooth device cannot be determined directly, since the detector only provides occupancy data, i.e. a Bluetooth device is located within a specific detection range. Nevertheless, it is possible to estimate the speed of a Bluetooth device indirectly. Furthermore, if the equipment rate of Bluetooth is not too low, an estimation of the vehicle counts is possible. Due to the mentioned facts, here, we concentrate on speed and vehicle number estimation on the basis of Bluetooth data.

Speed estimation

Generally, it is not possible to measure the motion of a Bluetooth device directly, since the Bluetooth inquiry process leads solely to the exchange of the device IDs of the communication partners (sender and receiver) and solely provides occupancy data [44]. However, since the timestamps and the detection ranges of the sender (the inquirer) are known we are capable of estimating the speed of a moved Bluetooth device, which can be obtained by computing the following equation:

$$V_{BT} \approx \frac{r_{BT}}{t_{Last} - t_{First}}, \quad t_{Last} - t_{First} > 0 \quad (1)$$

In equation (1) the estimated speed V_{BT} of the Bluetooth device is the result of the division of the detection range r_{BT} and the time difference of the first t_{First} and last detection t_{Last} of the Bluetooth device. Obviously, V_{BT} is more accurate the more frequent the same device is detected, which usually happens due to the periodicity of the inquiry process. That means, the less often a Bluetooth device is detected (“the quicker it moves through the detection range”), the higher is its speed and the bigger is the resulting estimation error. Vice-versa, the more often a Bluetooth device is detected within a specific detection range (“the slower it moves through the detection area”), the slower is its speed and the smaller is the resulting estimation error.

Clearly, in the case of a single Bluetooth detection, equation (1) cannot be calculated. However, in this case, a specific speed can be assumed by some heuristics, e.g.

$$V_{BT,1} = V_{max} \quad (2)$$

$$V_{BT,1} = \bar{V} \quad (3)$$

$$V_{BT,1} = f(V_{max}, \bar{V}, V_{BT}', \Delta V, \dots) \quad (4)$$

However, the assumption in equation (2) leads to smaller V_{BT} every time a new detection of the same Bluetooth device occurs yielding a valid velocity interval in which the true speed of the Bluetooth device is expected. Further, the application of equation (2) yields big errors in case of vehicles entering the detection area at low speeds. However, although this fact leads to systematic errors in the estimation of speeds by Bluetooth, equation (2) showed to be reasonable. Equation (3) may be adequate in case of generally low speeds or higher traffic volumes or in case of approaching a traffic light showing “red”. In equation (4) a general version of the estimation of V_{BT} is given, which includes other parameters that may affect the actual Bluetooth velocity. For instance, the speed difference of two successive vehicles ΔV describes the influence of the traffic state on V_{BT} , since $|\Delta V|$ and its variance will be different in heavy traffic in comparison to free flow. Further, the speed estimation V_{BT}' of a Bluetooth vehicle before may be similar to the current (unknown) speed due to quasi-steady state conditions of traffic flow within a small time interval. However, the dots in

equation (4) emphasise that other parameters should be considered here, depending on the intersection type, the time of day. Other influences that need to be considered and handled throughout the computation of equation (1) are:

- Not only vehicles, but pedestrians, cyclists, passengers of railways, trams, etc. may be equipped with Bluetooth (or even with more than one Bluetooth device in the case of busses), which would lead to a systematic error of V_{BT} . For this reason, a transport mode detection (sub-section 2.3.1.4) must be implemented before equation (1) is computed. Then, the estimation of the Bluetooth speed device can be considered only for vehicles approaching the intersection of interest.
- It is necessary to distinguish between vehicles approaching and leaving the intersection to avoid systematic speed errors. Therefore, Bluetooth devices leaving the intersection must be filtered (sub-section 2.3.1.5).

In Figure 17 the detection ranges of a centrally placed C2X-RSU and 4 directed Bluetooth detectors are shown qualitatively in case of an ideal installation. According to [40] and [43] such C2X-RSUs have a detection range from 100m up to 500m. Further, the following Bluetooth devices can be distinguished, which guarantee the mentioned ranges:

- class 1: transmission power 100mW, range 100m
- class 2: transmitting power 10mW, range 30m
- class 3: transmission power 1mW, range 10m

Class 1 receivers are rarely found in traffic. However, traffic participants mainly use Bluetooth devices comparable to either receiving class 3, e.g. Smartphones, hands-free communication devices, car audio, etc., or receiving class 2, e.g. laptops with Bluetooth sticks, printers, etc. In case the class of a Bluetooth device is known, equation (1) can be applied to obtain speed estimations for each arm of the intersection.

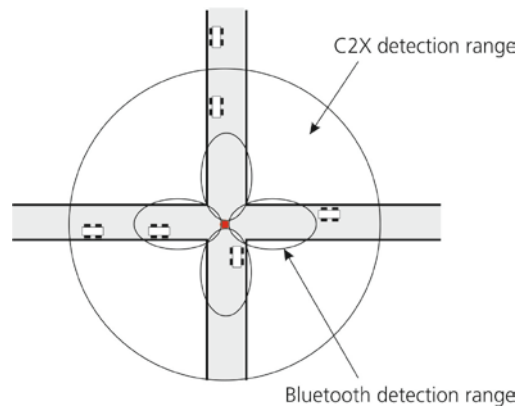


Figure 17: Idealised intersection scenario with detection ranges of a C2X-RSU and 4 Bluetooth detectors.

Vehicle count estimation

As mentioned above, the velocity estimation error is dependent on the velocities of the vehicles in the Bluetooth detection range of the inquirer. If a Bluetooth device remains within the detection range for long, i.e. it moves slowly, the detection of the device by the inquiry process is more probable. Vice-versa in case of a quick moving Bluetooth device, i.e. it remains within the detection area for a short time only, its detection is less likely. Consequently, there is an inverse proportionality between the probability detecting a Bluetooth device given a true speed and the speed (with $P(D|V)$ as conditional detection probability and speed V):

$$P(D|V) \sim V^{-1} \quad (5)$$

In Figure 18 the relationship of equation (5) is shown qualitatively.

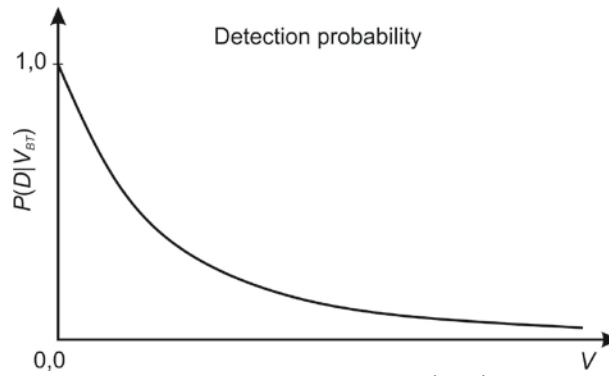


Figure 18: Qualitative curve of the detection probability $P(D|V)$ as a function of speed V .

As consequence of equation (5) in case there are several Bluetooth devices within the detection range, most of them are detected at low speeds and only a few of them are detected at high speeds. To benefit from the knowledge that every detection of a Bluetooth device corresponds to exactly one traffic participant, the a-priori probability density $P(D|V)$ can be estimated empirically and the number of vehicles can be estimated even in case of lower Bluetooth penetration ratios.

In case of low vehicle speeds and equally distributed Bluetooth devices in the road network as well as penetration ratios of approximately $(0.3;1]$ the number N of detected vehicles is (with the correction factor c):

$$N \approx \frac{\text{\#detected Bluetooth devices}}{\text{penetration rate}} \cdot c \quad (6)$$

which can be accumulated to obtain the estimated amount of vehicles that passed through the detection area. The application of equation (6) may fail under some circumstances, namely if

- the Bluetooth devices on the road network are not equally distributed,
- the equipment ratio is rather low, e.g. approximately less than 30%,
- the speeds of the Bluetooth devices are too high.

As a consequence, c is dependent on the traffic state, the speed distribution at the intersection of interest and needs to be determined carefully to obtain an acceptably accurate count of vehicles estimated by the detected Bluetooth device. In case of C2X the estimation of vehicle numbers is rather difficult, since the penetration ratio of C2X is extremely low. Applying equation (6) would result in intolerable high vehicles numbers then. Therefore, we currently resign vehicle number estimation in this respect.

2.3.1.2 Sensor data fusion of C2X and Bluetooth data

Clearly, speed and vehicle count information are erroneous and incomplete. Thus, to improve the speed estimation accuracy, a fusion method is needed, which is capable of coping with faulty and highly incomplete C2X and Bluetooth data. An adequate method to do so is the concept of Bayesian Networks (BN), which is capable of coping with incomplete data by the use of (conditional) probabilities. To get a detailed view about BN, their computation and application areas, learning procedures, etc. see e.g. [41] and [42].

Adopting BN for speed estimation requires the modelling of both, traffic process and detection (measuring) processes of the sensors as random processes. These processes are considered as nodes, whilst the cause and effect relationships among them are modelled as directed arcs, pointing from cause to effect. After modelling, the nodes are quantified by conditional probabilities representing the properties of the processes. It is necessary to know how the sensors determine the velocities of the Bluetooth devices taking into account the true velocities.

After that, the BN in question enables the handling and combination of the incomplete data of the sensors to estimate the speeds considering the properties of the sensors, the a-priori knowledge about the underlying traffic process and the measurement data.

BN for speed determination

In Figure 19 a “naïve” BN is shown that consists of the traffic process node V representing the physical instantaneous mean speed to be determined by C2X (sensor node V_C2X) or Bluetooth (sensor node V_BT). The arcs point from V to V_BT and V_C2X modelling the causal relationship between the physical speed and the measurement. The term $P(V)$ quantifies the a-priori knowledge, i.e. the expected speed probability, and the terms $P(V_{BT}|V)$ and $P(V_{C2X}|V)$ characterise the conditional probability densities (CPD) and thus, quantify the sensor properties. The sensor CPDs are also referred to as the sensor likelihoods. In case we know something at node V we can estimate what the sensors will measure probabilistically (causal evidences), which is important for learning the CPDs. In the case we have measured some data at V_BT and/or V_C2X we are able to find out what might have happened at node V (diagnostic evidences). This procedure is called inference.

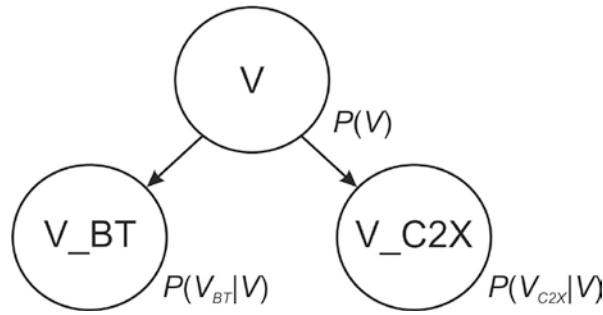


Figure 19: BN with the node V (physical instantaneous speed) and the sensor nodes V_BT and V_C2X (measured speeds). The conditional probability densities (CPDs) are given.

The joint probability distribution (JPD), which can be simply computed by the multiplication of the nodes' CPDs according to equation (7), characterises the probabilities taking into account the measurements at the nodes V_BT and/or V_C2X :

$$P(V, V_{BT}, V_{C2X}) = P(V) \cdot P(V_{BT}|V) \cdot P(V_{C2X}|V) \quad (7)$$

Clearly, equation (7) is only valid, if the Bluetooth (φ_{C2X}) and C2X data (φ_{C2X}) available are related to the same, i.e. overlapping, detection areas. If the detection area of Bluetooth is written as \mathcal{E}_{BT} and the detection area of C2X is written as \mathcal{E}_{C2X} , then equation (7) holds if

$$\varphi_{BT}, \varphi_{C2X} \in \mathcal{E}_{BT} \cap \mathcal{E}_{C2X} \quad (8)$$

In order to probabilistically estimate what is going on at node V , we compute equation (9), which – in this case – contains evidences about the measurements at the nodes V_BT and V_C2X . Consequently, the a-posteriori probability density $P(V|V_{BT}, V_{C2X})$, which probabilistically combines the measurements, is computed with the normalising constant $\alpha^{-1} = P(V_{BT}, V_{C2X})$ by the following fusion equation (α ensures that $\sum_i P(V_i|V_{BT}, V_{C2X}) = 1$) and taking into account equation (8):

$$P(V|V_{BT}, V_{C2X}) = \alpha \cdot P(V) \cdot P(V_{BT}|V) \cdot P(V_{C2X}|V) \quad (9)$$

Equation (9) is a simple (Bayesian) calculation rule realising the (weak) sensor fusion [38] of two data sources taking into account the expectance concerning the speed probability density (a-priori probability) $P(V)$ and the sensor likelihoods $P(V_{BT}|V)$ and $P(V_{C2X}|V)$ (sensor properties). However, the merging the measured C2X and/or Bluetooth data shall improve the accuracy and the reliability of the mean speed which is achieved due to sensor redundancy. On the other hand, due to the different detection ranges of Bluetooth and C2X technology it is possible to extend the detection horizon. Therefore, equation (9) needs to be applied carefully and in accordance with the applied

detection ranges, i.e. it is only valid, if the measurements are merged within the same detection ranges.

In case of only one data source is available providing speed data (equation (8) does not hold), equation (9) can be rewritten and computed as shown in equation (10) and equation (11):

$$\varphi_{BT} \in \Xi_{BT}; \varphi_{C2X} = \emptyset: P(V|V_{BT}) = \alpha_1 \cdot P(V) \cdot P(V_{BT}|V) \quad (10)$$

$$\varphi_{C2X} \in \Xi_{C2X}; \varphi_{BT} = \emptyset: P(V|V_{C2X}) = \alpha_2 \cdot P(V) \cdot P(V_{C2X}|V) \quad (11)$$

In addition, another version of equation (9) can be formulated in the case data of both detection technologies are available, but are not overlapping, i.e.:

$$\varphi_{BT}, \varphi_{BT} \notin \Xi_{BT} \cap \Xi_{C2X} \quad (12)$$

In case equation (12) holds, the fusion process formulated by equation (9) is usually wrong, except taking into account specific requirements to the traffic process in general. Due to this fact, the fusion process according to equation (12) is not considered throughout this investigation. Thus, in this paper we concentrate on the fusion process according to the equations (8) - (11).

Then, the most likely probability velocity of the vehicle on the street can be obtained by the application of an adequate estimator. In [52] different estimators are described to get a deeper impression. Here, we applied the maximum a-posteriori estimator (MAP), which always yields the velocity for which the a-posteriori distribution $P(V|V_{BT}, V_{C2X})$ has its maximum:

$$\hat{V} = \arg \max_V P(V|V_{BT}, V_{C2X}) \quad (13)$$

However, it may happen that $P(V|V_{BT}, V_{C2X})$ is a zero a-posteriori probability distribution, which makes it impossible to obtain the estimate \hat{V} . This may happen, if the sensor likelihoods of the Bluetooth and the C2X sensor do not “overlap”, which is due to the learning procedure of the likelihoods and will be described later. Then, the equations (9) and (10) will be applied to obtain separate speed estimations for Bluetooth \hat{V}_{BT} and C2X \hat{V}_{C2X} , which are combined afterwards. Here, we applied a simple weighted mean value operator of both estimations (with the weight w and $0 \leq w \leq 1$) to obtain a good estimation:

$$\hat{V} \approx w \cdot \hat{V}_{BT} + (1 - w) \cdot \hat{V}_{C2X} \quad (14)$$

Obviously, w must be chosen carefully, e.g. the result of the more accurate sensor can be weighted higher than the less accurate sensor.

Due to the different detection ranges of the competing wireless technologies altogether 5 (6) fusion cases can occur:

1. Bluetooth only
2. C2X only
3. Bluetooth and C2X (i.e. competing sensors in the same detection area)
 - 3.1 Non-overlapping sensor likelihoods (specific case of 3.)
4. Bluetooth and C2X for different ranges (i.e. competing sensors for overlapping detection areas)
5. No data

Extension of the BN

As mentioned in section 2.3.1.1 and expressed by equation (4) different factors may influence the determination of the particular Bluetooth detector to obtain a good estimate \hat{V} by data fusion. In the following, some of the affecting factors are discussed and as consequently the BN in Figure 19 is extended to the BN shown in Figure 20. There, we consider black nodes (real BN nodes) and grey nodes (important for clarity reasons, but not needed for processing):

- The speed of the vehicles (node V) is clearly affected by the traffic density (and sometimes even vice-versa) (node D). To put it simple this means that the speed is low if the density is high and vice-versa. Further, the speed of the vehicles is clearly affected by the traffic light control (node TLC) yielding to more or less stoppages and waiting times and thus different speeds and means speeds. Therefore, there are directed arcs from D and TLC to V.
- A Bluetooth detector is originally an occupancy detector (node O), which is capable of counting Bluetooth devices, and not a speed detector. Since, the estimation of the speed (node V_BT) can be realised there, node O can be neglected when processing the BN.
- To obtain a good estimate of the speed of a Bluetooth device (node V_BT), it is assumed that the current Bluetooth speed is dependent on the speed of the preceding Bluetooth device (node V_BT'). This is based on the steady-state assumption of traffic flow taking into account that the current traffic state does not usually change heavily in a relatively short amount of time. In a similar manner we determine the speed difference (node ΔV) between the current and the preceding Bluetooth device. Therefore, there are directed arcs from O, ΔV and V_BT' to V_BT.
- Finally, no claim is made for completeness, since the BN in question only explains a section of the reality. Our future work will deal with extending it further.

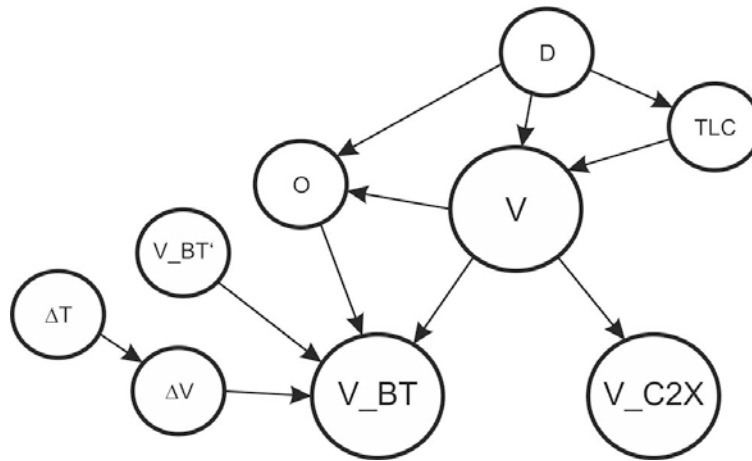


Figure 20. Extended BN including some of the dependencies for a reliable determination of the velocity.

For this deliverable we did a first investigation of the fusion process on the basis of BN. Therefore, the BN in Figure 20 was simplified to the BN shown in Figure 21, which takes the nodes V_BT' and ΔV into account.

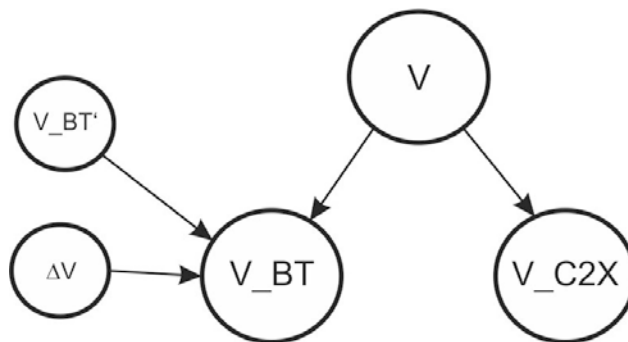


Figure 21. Simplified BN according to Figure 20 for analysing the fusion process

Analogous to section 0 we can formulate the equations needed for understanding and processing the BN shown in Figure 21 taking into account the additional nodes to model the affection of node V_BT. The JPD is then:

$$P(V, V_{BT}, V_{C2X}, \Delta V, V_{BT}') = P(V) \cdot P(V_{BT}|V, \Delta V, V_{BT}') \cdot P(V_{C2X}|V) \cdot P(\Delta V) \cdot P(V_{BT}') \quad (15)$$

The a-posteriori probability density $P(V|V_{BT}, V_{C2X}, \Delta V, V_{BT}')$, which considers the diagnostic evidences obtained in V_{BT} and V_{C2X} as well as the causal evidences in V_{BT}' and ΔV is computed with the normalising constant α by the following fusion equation (α ensures that $\sum_i P(V_i|V_{BT}, V_{C2X}, \Delta V, V_{BT}') = 1$):

$$\begin{aligned} P(V|V_{BT}, V_{C2X}, \Delta V, V_{BT}') \\ = \alpha \cdot P(V) \cdot P(V_{BT}|V, \Delta V, V_{BT}') \cdot P(V_{C2X}|V) \cdot P(\Delta V) \cdot P(V_{BT}') \end{aligned} \quad (16)$$

2.3.1.3 Learning

Reliable and accurate speed estimations according to the equations (9) – (16) are only possible if the sensor likelihoods $P(V_{BT}|V, \Delta V, V_{BT}')$ and $P(V_{C2X}|V)$ as well as the a-priori probability densities $P(V)$, $P(\Delta V)$ and $P(V_{BT}')$ are modelled and quantified accurately.

A-priori probabilities

Generally, an a-priori distribution represents the statistical knowledge of the underlying process. Any a-priori distribution can be modelled as a uniform distribution in case nothing is known about the underlying process, but this usually yields to some realisations of V which are over- some other are under estimated. Otherwise, the a-priori-distribution in question can be modelled and quantified by accurate reference measurements or by expert knowledge.

The process of learning static or even adaptive a-priori probability densities is known and well-studied, which can be found in the literature, e.g. [41], [53] and [54]. Here, we focus on static learning.

Here, the these a-priori distributions are the traffic process with the instantaneous speeds V , the difference speed ΔV to the preceding vehicle and the detected mean speed of the preceding vehicle V_{BT}' .

a) a-priori speed distribution $P(V)$

The prior $P(V)$ represents the statistical knowledge about the expected physical speed of the moving vehicles within the detection range. It is reasonable to determine $P(V)$ by reference measurements of a highly accurate sensor. Estimating $P(V)$ requires knowledge of the environment, road network and road type, traffic control, time of the day, etc. Here, we concentrate on an unaffected traffic process according to the BN in Figure 21.

Due to the fact that two different wireless communication technologies with different detection ranges are considered here, also two a-priori probability density functions for the different detection ranges ought to be determined. But, since we consider the same detection range for the fusion process of both sensors, only one a-priori speed distribution is necessary.

b) a-priori speed difference distribution $P(\Delta V)$

The prior $P(\Delta V)$ represents the statistical knowledge about the detected difference of the current Bluetooth device and the mean speed of the preceding Bluetooth device when leaving the detection area. Clearly, the speed difference is related to a vehicle some time interval ΔT earlier. Thus, $P(\Delta V)$ can only be determined taking into account the preceding measurements of the speed differences.

c) a-priori detected mean speed distribution of the preceding Bluetooth device $P(V_{BT}')$

Analogous to the determination of the a-priori speed difference distribution the prior $P(V_{BT}')$ represents the statistical knowledge of the detected mean speed of the preceding Bluetooth device leaving the detection area of the Bluetooth detector. Again, $P(V_{BT}')$ can only be obtained taking into account the measurements of the preceding detected mean speeds of the Bluetooth device.

Sensor likelihoods

The sensor likelihoods quantify the properties of the sensors as conditional probabilities, i.e. the CPDs $P(V_{BT}|V, \Delta V, V_{BT}')$ and $P(V_{C2X}|V)$ in the case of Bluetooth and C2X respectively. The performances of the sensors are dependent on their functional principle, the surrounding environment and other phenomena, such as thermal noise, etc. Since we can assume that a specific vehicle somehow behaves like its preceding vehicle taking into account the current traffic conditions it is reasonable that there is a correlation of the speeds and the speed differences too. Therefore, we take the sensor's functional principles as well as the affection of the Bluetooth based speed detection by the speed difference and the detected mean speed of the preceding Bluetooth devices into account without consideration of the environment.

The properties of the sensors of interest can be learnt by more accurate and reliable sensors and/or by fused sensors that provide data with the desired accuracy. Due to the fact that we consider only unaffected sensors in their environment, the learning of the sensor CPDs is characterised by simple histograms with regard to the true physical speeds and the mentioned influence parameters ΔV and V_{BT}' . The process of learning the CPDs of the sensors is a well-studied process, which can be found detailed in the literature, e.g. [41], and will not be discussed here.

Conclusions for learning

Due to the mentioned conditions for learning the a-priori probability density distributions of the parameters of interest as well as the likelihoods of the Bluetooth and C2X sensors the following consequences need to be stated:

- The Bayesian data fusion method introduced in sub-section 2.3.1.2 is only as good as the a-priori probabilities model the statistical characteristics of the underlying traffic process are and as good the sensor likelihoods model the properties of the sensors considered for data fusion.
- The more dimensions the considered sensor likelihoods have the more samples for learning are needed. Due to the fact that the BN according to Figure 21 the CPD for the Bluetooth sensor considers the parameters ΔV and V_{BT}' are based on the mean speed of a detected Bluetooth device, the joint sensor likelihoods of the Bluetooth and the C2X will offer non-overlapping probabilities, i.e. the fusion equation (16) will sometimes yield zero a-posteriori probability distributions, which need to be handled by another method. As mentioned above in this case equation (14) will be applied.

2.3.1.4 Transport mode detection

Due to the fact that Bluetooth devices are available in almost every transport mode, e.g. Smartphones (pedestrians, all vehicle classes, bicycles), hands-free communication systems (all motorised vehicles), systematic errors for vehicle counts and the estimated speeds would occur. Imagine a crowd in front of a museum near an intersection using their Smartphones to get information about the exhibits there via Bluetooth: the number of detected vehicles would be overestimated, whereas the speeds would be underestimated by far. Therefore, a transport mode detection is required to avoid or at least mitigate the systematic errors. This is current state of research, e.g. in[45], whereas, however, no method is currently known, which realises a transport mode detection on the basis of Bluetooth device data, i.e. Bluetooth MAC addresses or their encrypted versions. Since not every intersection of a road network is frequented by all transport modes (pedestrians, cyclists, road and rail traffic), e.g. on express and motorways, transport mode detection can be adapted to the modes available. Therefore, in this study transport mode detection is addressed, which will be part of our future work.

2.3.1.5 Direction filtering

Bluetooth detection can be realised with omnidirectional or directional antennae. Due to the lack of positional information when detecting a Bluetooth device it is sometimes necessary to restrict the positional and angular arbitrariness of the device by the use of directional antennae. In case of an intersection with 4 arms it is essentially important to know on which arm a vehicle approaches the intersection and on which arm it leaves the intersection. Therefore, it is reasonable to install as many directed Bluetooth detectors as intersection arms available, i.e. an intersection with 4 arms should be equipped with 4 directed Bluetooth detectors. Then, only one of the detectors is capable of detecting Bluetooth devices of one specific arm, whereas it is “blind” against the other arms.

However, although there can be found trials to identify approaching and leaving vehicles by RSSI (Received Signal Strength Indicator) value in literature, it is not satisfactorily possible to achieve this goal due to the extreme deviation of the RSSI values[48]. Furthermore, in the present state of research it is not (yet) possible, even with directed antennae, to estimate the position of a Bluetooth device on a specific lane of a road.

2.3.2 Experimental Results

The method proposed in chapter 2.3.1 was implemented and tested using the microscopic traffic simulation SUMO (Simulation of Urban MObility), which has been further developed and applied within several EU projects, e.g. DRIVE C2X, iTetris and COLOMBO; see the literature, e.g. [46], [55] and [56], for more details. In this section the experimental setup (sub-section 2.3.2.1) and the simulation results (sub-section 2.3.2.2) are presented.

2.3.2.1 Experimental Setup

SUMO Traffic Simulation Setup

In Figure 22 the simulated scenario of a signalised intersection with 4 arms, which strongly relates to the idealised intersection in Figure 17, can be seen. Here, the boundaries of the 4 directed Bluetooth detection ranges are marked as green lines (entering vehicles) and red lines near the stopping line (leaving vehicles). For clarity reasons, the detection range of the C2X-RSU is not shown here. A RSU, which is capable of collecting C2X messages of the vehicles, was integrated into the RiLSA 1 scenario in SUMO [47]. The detection range was set to $r_{C2X} = 200\text{m}$.

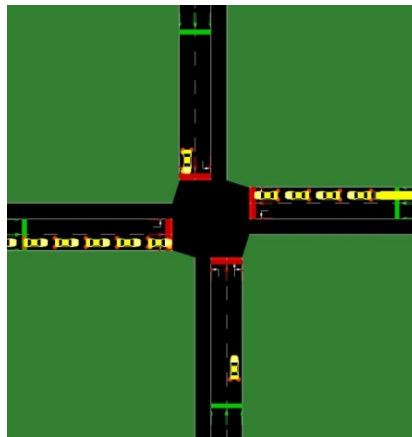


Figure 22: Simulated intersection with 4 arms (COLOMBO RiLSA 1 example[47]).

Additionally, the RSU is equipped with 4 standard Bluetooth detectors, which scan the 4 arms of the intersection with directed antennae separately. This ensures that the RSU can differ between the intersection arms. As Bluetooth detectors we considered standard class 2 receivers assuming an maximum detection range of $r_{BT} = 30\text{m}$.

In the simulation we retrieved vehicle counting and speed estimation in dependence on the different C2X penetration rates, whereas the Bluetooth penetration rate remained constant at 30%. The C2X penetration rates are [1; 2; 5; 10; 20; 50] %.

Further parameters, which need to be introduced:

- Simulation time: 3,695.2 s (36,953 simulation steps)
- Simulation runs:
 - o In case of learning the a-priori probabilities and sensor likelihoods: 400
 - o In case of data fusion: 10
- Simulation step size: 0.1 s
- Determining vehicle counts:
 - o In the case of an urban signalised intersection network we were capable of estimating the vehicle count for speeds up to 50 km/h (13.9 m/s) satisfactorily.
 - o Since no analysis of the speed and traffic state dependence of the correction factor c has been done yet, it was set for the four intersection arms differently considering the traffic specific situations, i.e.:
 - $c_{\text{North}} = 1.21$
 - $c_{\text{East}} = 1.72$
 - $c_{\text{South}} = 1.0$
 - $c_{\text{West}} = 1.74$
- Basis detection probability
 - o Bluetooth:
 - The inquiry process was modelled probabilistically, i.e. it was assumed that within the simulation step size the instantaneous detection probability is about 1.7%, which leads to a detection probability of a Bluetooth device of approximately 70% within 2.2 seconds. This was found by own investigations and is in accordance with the Bluetooth detection behaviour.
 - We assumed that a Bluetooth device cannot be detected twice or more often within 2.56s after first detection, which is related to the inquiry scan of one of the two frequency trains.
 - o C2X: it was assumed that 90% of the C2X messages sent by the vehicles are received by the C2X-RSU

TSE Application Setup

A BN according to Figure 21 was created and its a-priori probability density functions as well as the sensor likelihoods were learnt for integer velocities from 0 to 19 m/s and speed difference values from -20 to 20 m/s. Although the detection ranges of the Bluetooth and the C2X detector are different, here, the fusion process is applied for overlapping detection ranges (see equation (8)).

a) A-priori probabilities

In Figure 23 the learnt a-priori probability densities $P(V)$ are shown for the northern (left picture) and the western arms (right picture). It can be stated that the $P(V)$ behave as expected: We can see that there are mostly two peaks, one at low (0 m/s) and one at high speeds (14 m/s), which is due to the fact that in this intersection scenario, either waiting due to traffic light phase red, or free flow with the nominal speed of 13.9 m/s due to green are the two dominant speeds. Particularly in the Northern arm the low speeds are dominating, and in the Western arm both speeds are more or less equally likely.

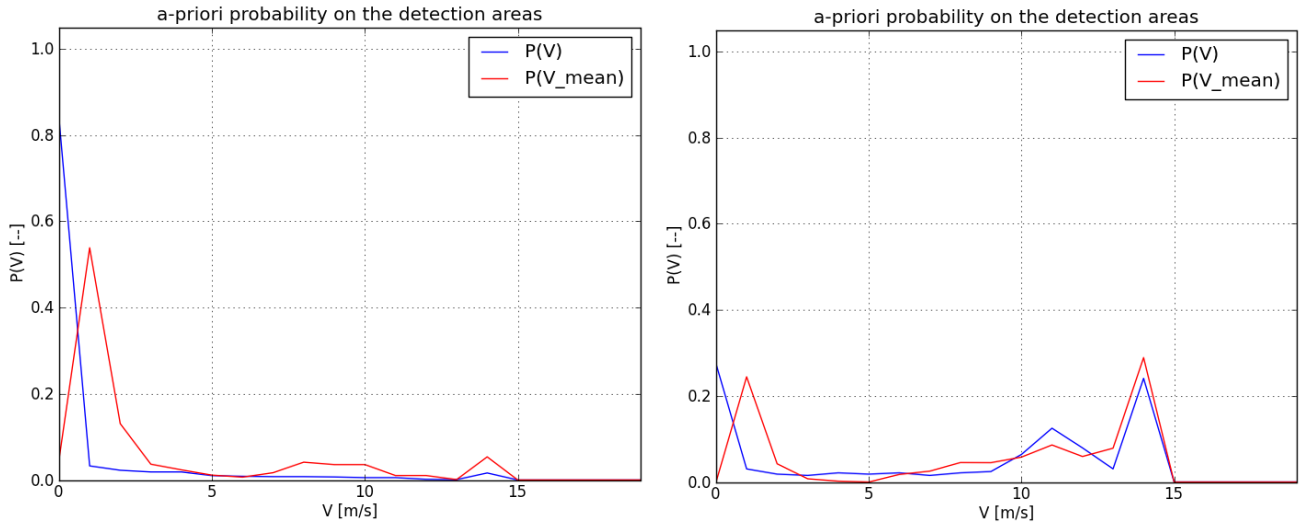


Figure 23: A-priori probability densities $P(V)$ of the northern (left) and western arm (right) of the intersection. Additionally, the a-priori distribution density for the mean speed is shown.

In Figure 24 the a-priori probability densities $P(\Delta V)$ of the northern and western arms, respectively, are shown. It can be seen, that there is a well-defined peak for $\Delta V = 0$ m/s for an almost symmetric density function. The density of the northern arm is much more peaked than the one of the western arm, which emphasises the different traffic states for the western arm.

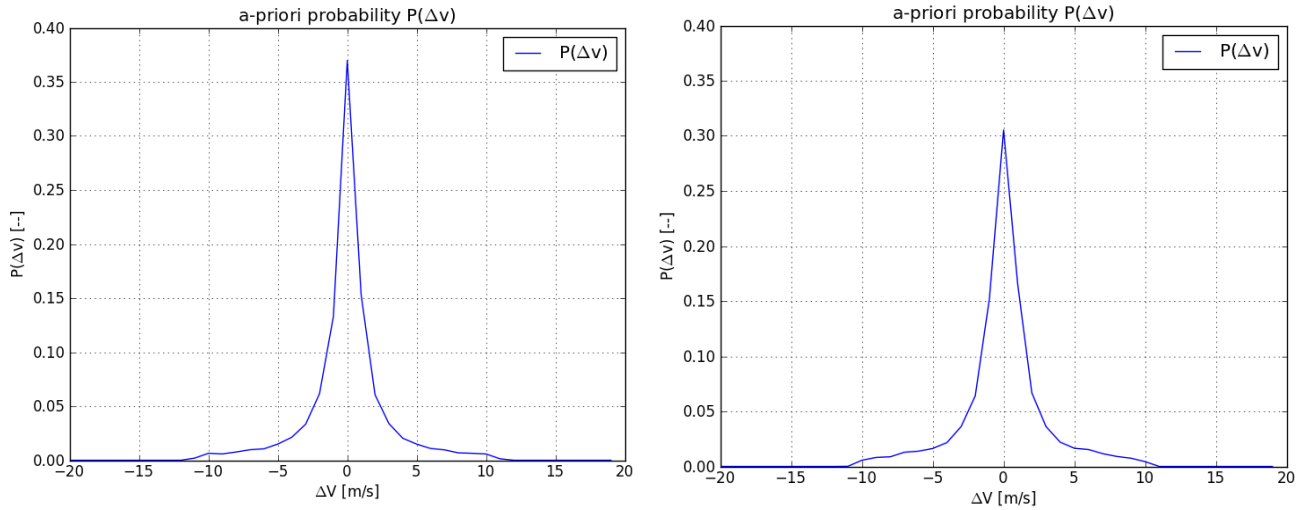


Figure 24. A-priori probability densities $P(\Delta V)$ of the northern (left) and western (right) arms of the intersection.

In Figure 25 the a-priori probability density $P(V_{BT}')$ of the northern and western arms, respectively, are shown. It can be seen, that the mean speed of the detected Bluetooth device, which entered, passed through and left the detection area has its maximum at 1 m/s for the Northern and 2 m/s for the western arm. In both cases the probability densities decrease quickly.

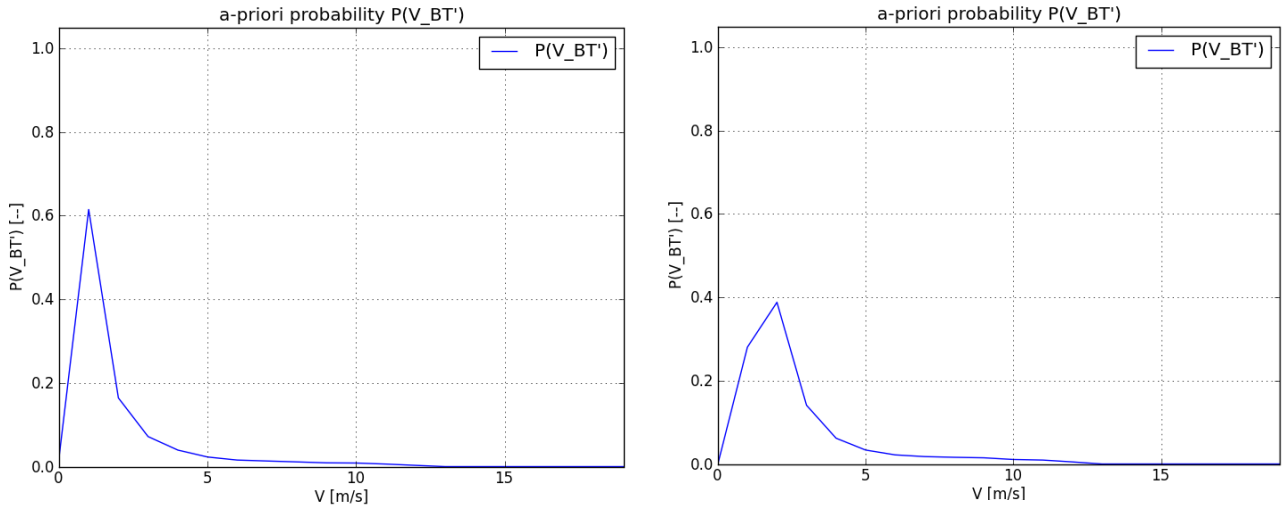
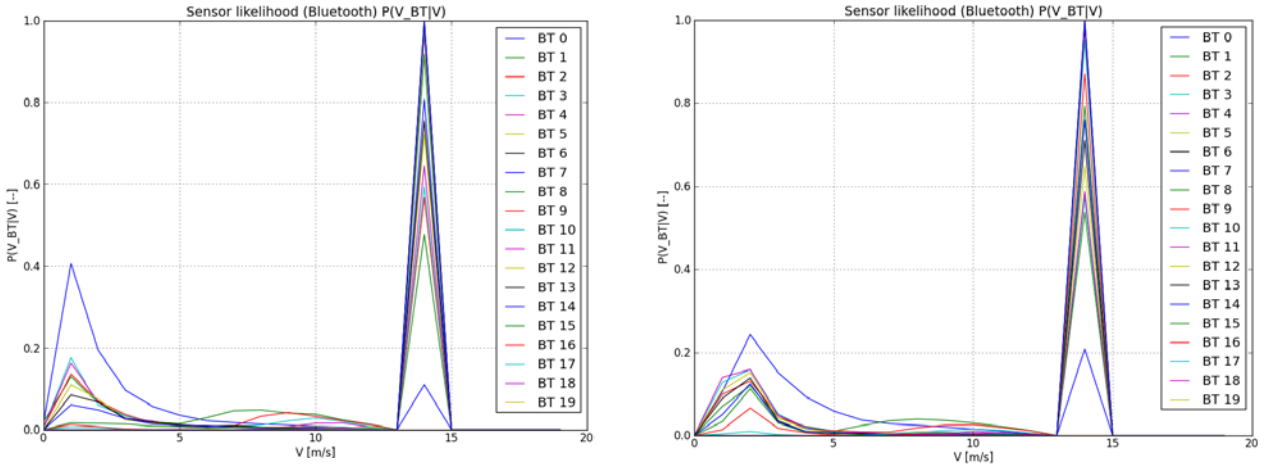


Figure 25. A-priori probability densities $P(V_{BT}')$ of the northern (left) and western (right) arms of the intersection.

b) Sensor likelihoods

In the following some examples for the learnt sensor likelihoods for speed estimation are given. The sensor likelihood for the Bluetooth detector, i.e. $P(V_{BT}|V, \Delta V, V_{BT}')$, cannot be visualised in a reasonable manner due to its high dimensionality. Therefore, the likelihood is processed to obtain the marginalised version of it, which is simply $P(V_{BT}|V)$. Both, $P(V_{BT}|V)$ for Bluetooth and, $P(V_{C2X}|V)$ for C2X are shown for the northern (left side) and the western (right side) intersection arms. There, the sensor CPDs (y-axis) are plotted for different detected speeds (x-axis) given the true physical speeds discretised as integers from 0 to 19 m/s, which is shown by the legends, e.g. BT 15 means the true speed of 15 m/s for Bluetooth.



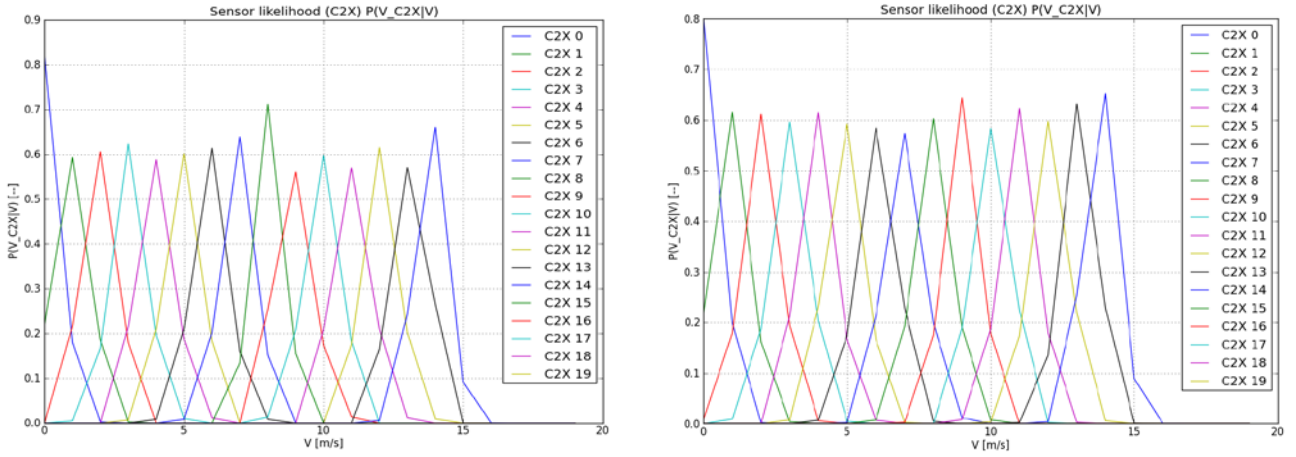


Figure 26: CPDs of the sensor likelihoods $P(V_{BT}|V)$ for marginalised Bluetooth (top row) and $P(V_{C2X}|V)$ for C2X (bottom row) for the northern (left) and western (right) intersection arms.

It can be stated that in case of C2X the speed estimation is as expected, i.e. the estimated values spread around the true physical speed, which is due to the fact that the positional and speed errors were modelled according to normal distributions (not shown here). In case of Bluetooth the situation is more complex. Due to the fact that a Bluetooth detection is more likely in case of low speeds than at high speeds the Bluetooth sensor likelihood shows the expected behaviour for low speeds. But, particularly, in case of high speeds at 14 m/s we see that a Bluetooth detector provides frequent speed overestimations, which is obviously due to the fact of applying equation (2) in case of the first detection of a Bluetooth device entering the detection area. Moderate speeds are – like low speeds – estimated as expected according to the a-priori probability of the speed in Figure 23. Consequently, particularly low and high speeds are to be expected by the Bluetooth detector. Thanks to modelling the Bluetooth sensor likelihood by two additional nodes (Figure 21) we are capable of improving the speed estimations of the Bluetooth detector.

c) TSE data fusion cases

As mentioned at the end of sub-section 2.3.1.2 there are altogether 6 cases, which had to be considered for data fusion. In this investigation we implemented 4 of them, i.e. the cases “Bluetooth only”, “C2X only”, “Bluetooth and C2X”, “Bluetooth and C2X non-overlapping”. The cases “Bluetooth and C2X for different detection areas” and “no data” are not considered here, but will be part of our future work.

d) Transport mode detection

As mentioned in sub-section 2.3.1.4 the transport mode detection is part of current research. Here, in this study it is assumed that such a method is available.

e) Direction filtering

Distinguishing between approaching and leaving an intersection (sub-section 2.3.1.5) is realised by combining the detection results of the several Bluetooth detectors. If a Bluetooth device passes a detector and it was detected at another detector before, it is evident that it leaves the intersection, otherwise it approaches the intersection. Clearly, an adequate algorithm for direction filtering is required, but this approach is satisfactory for this feasibility study.

Evaluation setup

To evaluate the fusion of Bluetooth and C2X data for different C2X penetration rates by the use of the BN modelled in sub-section 2.3.1.2 the following quality parameters were determined:

- Mean root mean square error v_{RMS} between the true and the estimated speeds indicating the accuracy of the fusion method

- Mean maximum vehicle count error $\#_{|e|}$ between the true and the measured vehicle counts indicating the accuracy of the vehicle count method
- Mean completeness q_c identify how complete the speed data are

Additionally, the histograms of the absolute speed error $v_{||}$ between the true and the estimated speeds are also computed.

2.3.2.2 Simulation results

In this sub-section the fusion results for different C2X penetration rates are determined according to the requirements met in the simulation, TSE and evaluation setups according to sub-section 2.3.2.1. Therefore, the correction factor c was set to reasonable values and the quality parameters v_{RMS} , $\#_{|e|}$ and q_c are computed for each intersection arm separately.

In

Table 2.8 the mean RMS v_{RMS} for the speed determination for different C2X penetration rates is shown. In case of the eastern and western arms v_{RMS} decreases as expected from 4.5 to 2.3 and from 5.0 to 2.4 m/s, respectively. Due to the higher accuracy of the more frequent C2X speed estimations the maximum of the a-posteriori probability distribution is shifted to more accurate values since the contribution of the less accurate Bluetooth detector decreases.

It should be noted that v_{RMS} is almost constant (it even slightly increases) for the northern and southern intersection arms for all penetration rates at about 2.2 to 2.4 and 1.8 to 2.1 m/s, respectively. This behaviour can be explained due to the fact that the vehicles statistically stick to the lower speeds, which is modelled by the a-priori probability. Therefore, independently on the data the Bluetooth sensor as well as the C2X detector provide, the maximum of the a-posteriori probability distribution $P(V|V_{BT}, V_{C2X}, \Delta V, V_{BT}')$ provides low speed values, which is statistically true. The slight increase of the v_{RMS} in case of higher C2X penetration rates (although the C2X detector is more accurate than the Bluetooth sensor) is also a resulting effect. In case of very high C2X penetration ratios the massive presence of accurate C2X data enables the displacement of the maximum of $P(V|V_{BT}, V_{C2X}, \Delta V, V_{BT}')$ to increase the overall accuracy again.

Table 2.8. Mean root mean square error v_{RMS} of the speed determination.

	v_{RMS} [m/s]			
C2X penetration rate	northern arm	eastern arm	southern arm	western arm
1%	2.3	4.5	1.8	5.0
2%	2.2	4.4	1.9	4.7
5%	2.3	4.0	1.9	4.4
10%	2.4	3.5	2.1	3.9
20%	2.4	3.1	2.1	3.3
50%	2.2	2.3	2.1	2.4

In Figure 27 and Figure 28 the histograms of the absolute speed error $v_{||}$ for the C2X penetration rates of 1% and 50% are shown for the northern and southern as well as the Eastern and western intersection arms, respectively. Particularly in Figure 28 it can be seen that big speed errors for the eastern and western arms occur more often than for the northern and eastern arms. Further, in case

of strongly increased C2X penetration rates of 50% these high speed difference errors are highly decreased.

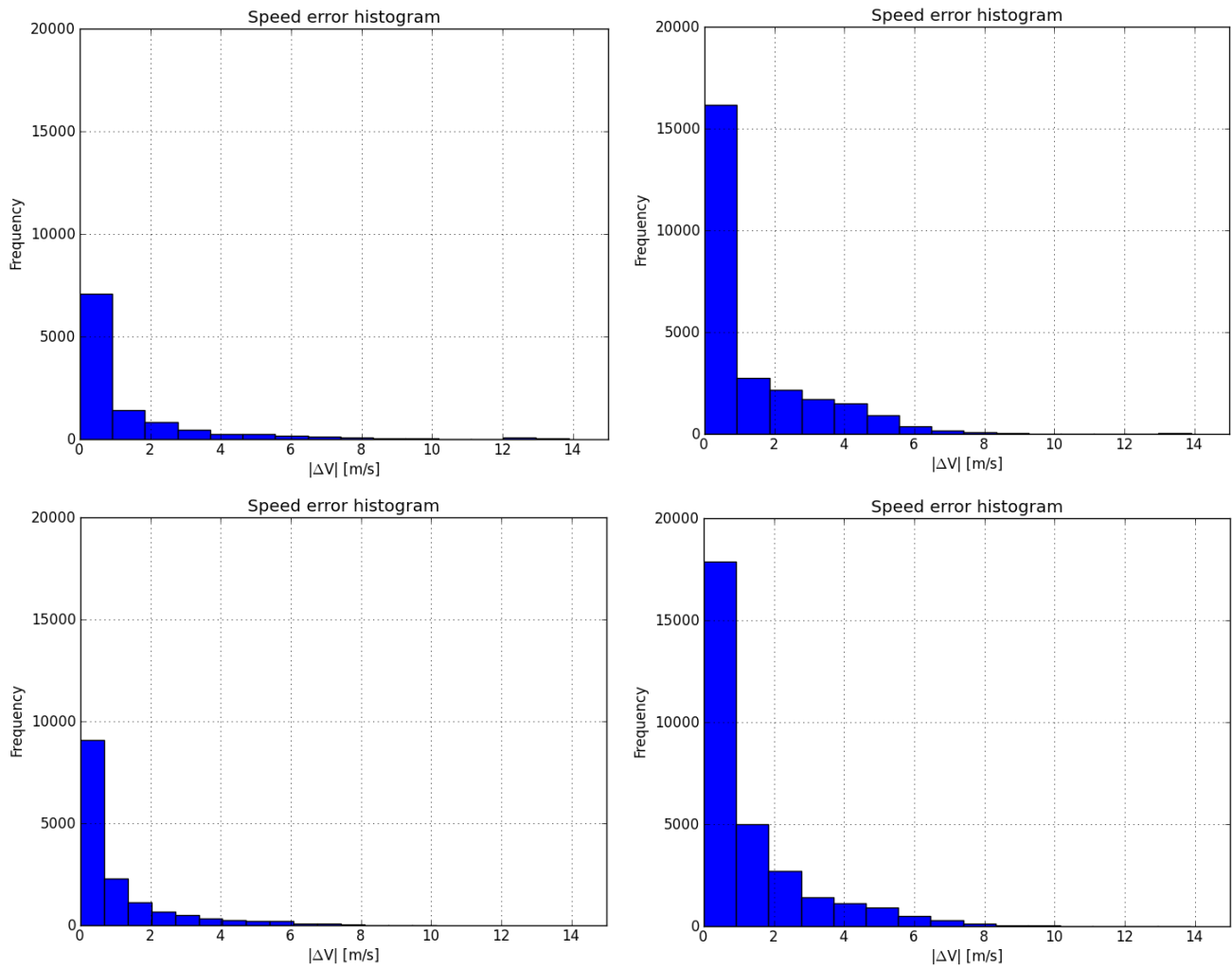


Figure 27. Histograms of absolute speed error for the northern (top row) and the southern arm (bottom row) for the C2X penetration rates 1% (left column) and 50% (right column).

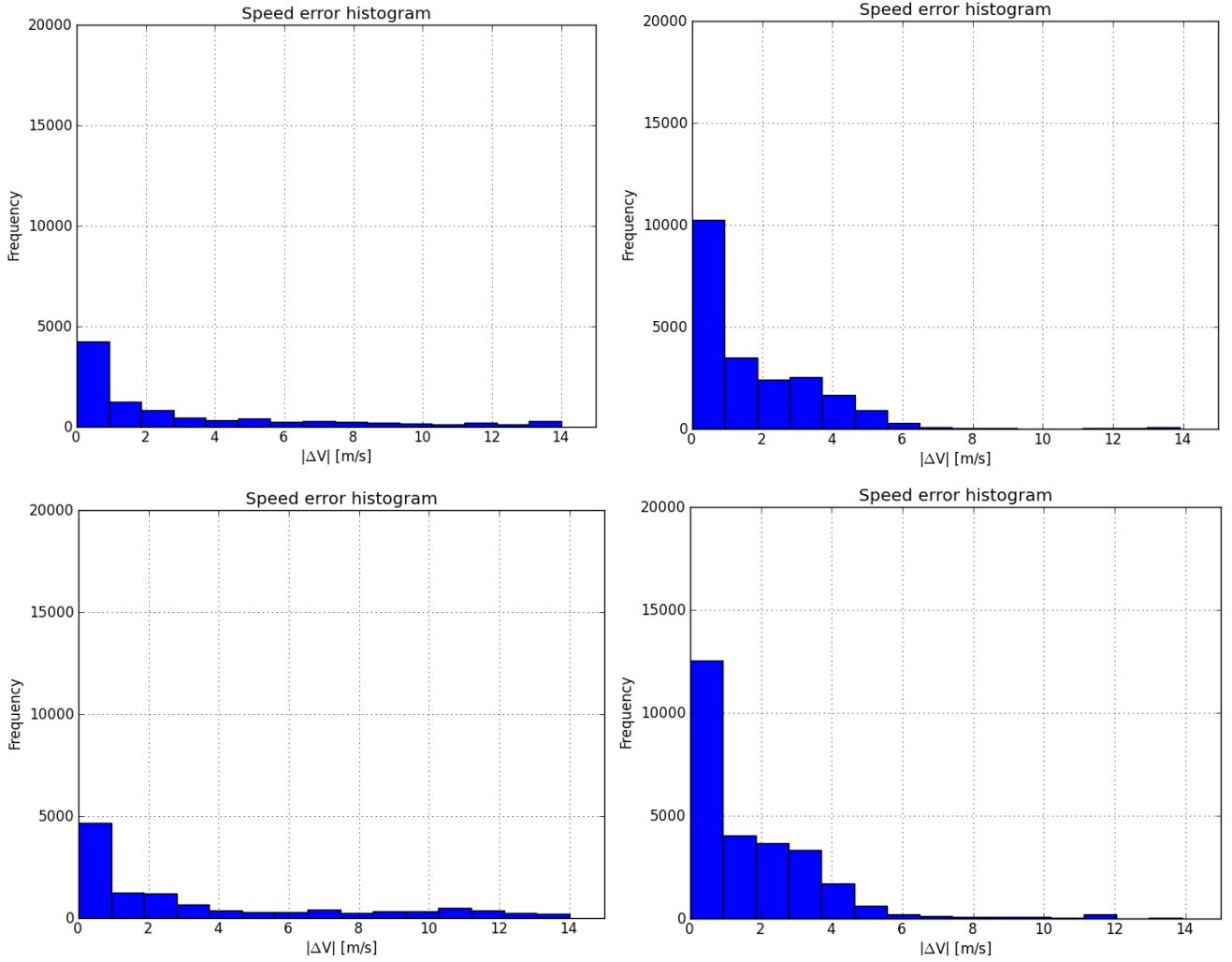


Figure 28. Histograms of absolute speed error for the eastern (top row) and the western arm (bottom row) for the C2X penetration rates 1% (left column) and 50% (right column).

The vehicle count error is independent of the C2X equipment rate, since it is determined by the Bluetooth occupancy detection only. In our experiments we obtained the maximum vehicle count error for the intersection arms in the following intervals:

- Northern arm: $\#_{|e|} = [24; 33]$ veh, which is equal to $[8.7; 11.7] \%$
- Eastern arm: $\#_{|e|} = [20; 37]$ veh, which is equal to $[3.0; 5.5] \%$
- Southern arm: $\#_{|e|} = [7; 13]$ veh, which is equal to $[2.5; 4.4] \%$
- Western arm: $\#_{|e|} = [24; 45]$ veh, which is equal to $[2.7; 4.9] \%$

It can be stated that the maximum vehicle count error is about 12% for the northern arm, whereas for all other arms it is less than 6%, which can be – also for traffic and transportation management – quite satisfactory.

In Figure 29 four examples of vehicle count errors $\#_e$ over time are shown for the different intersection arms. In these examples it can be seen how $\#_{|e|}$ fluctuates over time. If $\#_e > 0$, there is a vehicle count overestimation, while for $\#_e < 0$ the vehicle counts are underestimated.

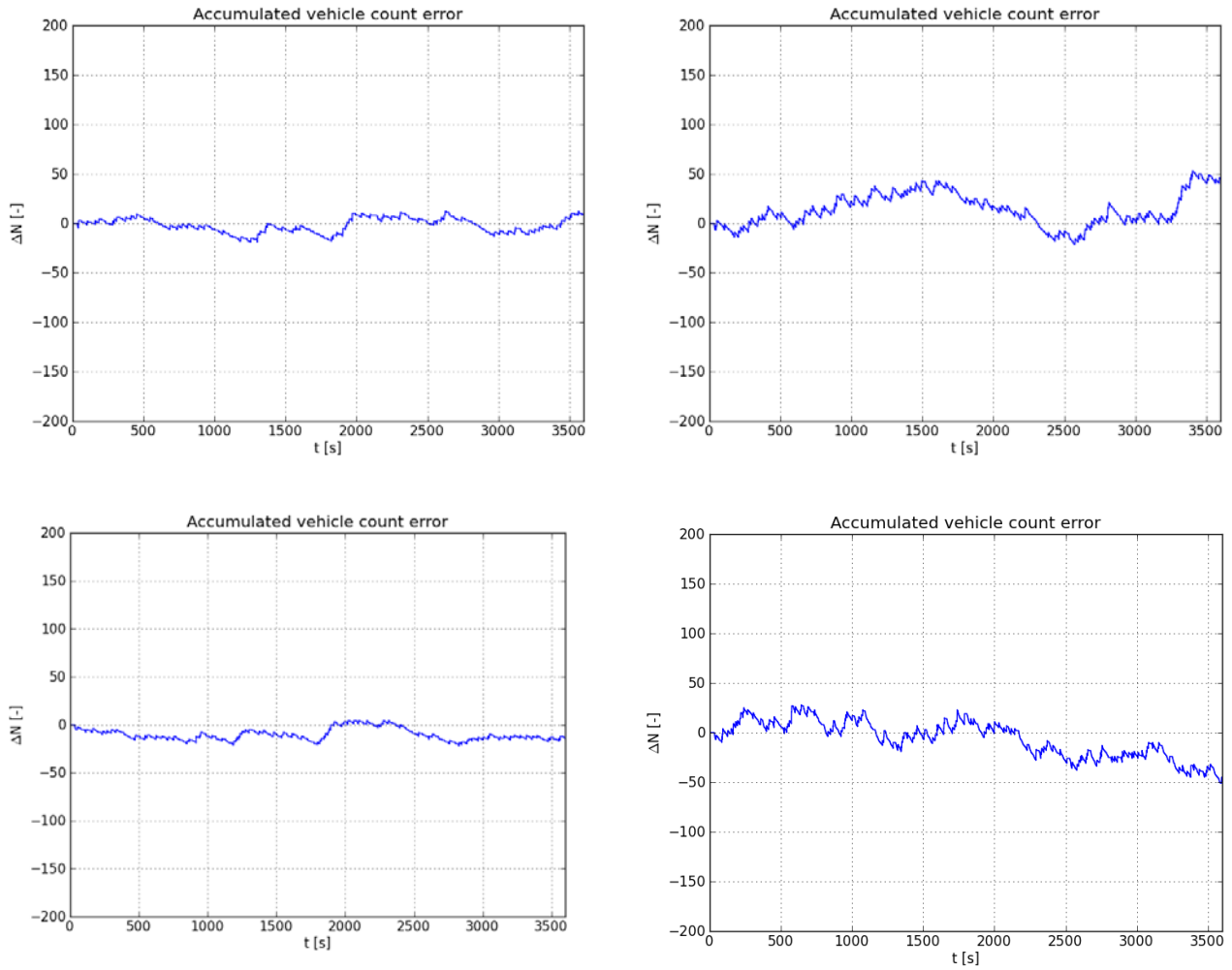


Figure 29. Vehicle count error $\#_e$ for the northern (top left), eastern (top right), southern (bottom left) and western (bottom right) intersection arms.

In Table 2.9 the mean completeness q_c of the speed determination is given. It can be stated that due to data fusion we are able to increase the detection horizon from approximately 36% in case of 1% C2X penetration rate to more than 90% in case of 50% C2X penetration rate.

Table 2.9. Mean completeness q_c of the speed determination.

C2X penetration ratio	v_{RMS} [m/s]			
	northern arm	eastern arm	southern arm	western arm
1%	37.6	36.2	40.6	40.7
2%	39.2	36.1	42.8	42.7
5%	42.6	44.2	48.6	50.0
10%	50.1	53.5	58.0	56.8
20%	60.8	64.6	67.0	71.9
50%	86.2	87.3	88.1	92.0

2.3.3 Discussions & Future Prospects

In COLOMBO an algorithm was developed and tested, which is capable of determining vehicle counts and capable of fusing rare C2X speed and moderately frequent Bluetooth occupancy data. It could be shown that the detection horizon and the overall accuracy of the speed estimation were increased.

Although a Bluetooth detector is originally unable to provide speed estimation information of Bluetooth devices passing its detection area, it is possible to do so if the detection range of the Bluetooth detector and the time interval of the vehicles passing are known. For this we modelled the traffic and measuring processes as a Bayesian Network and applied it for accurate and reliable speed estimation.

We analysed the performance of algorithm by computing some quality measures and obtained an RMS for speed estimation in case of a C2X penetration rate of only 1% between approximately 2 and 5 m/s. Since the allowed speed on the simulation was 13.9 m/s even the worst results achieved are even more than twice as well as throwing dices. Since the investigations were made on basis of traffic simulations and other assumptions (e.g. constant penetration rate for Bluetooth of 30%, uniformly distributed C2X and Bluetooth technology, just one analysed urban intersection, etc.) we expect the results worse. However, the obtained results show that even for C2X penetration rates of 1% we are able to achieve accurate and reliable speed data for traffic and transportation management purposes. There seems to be a high potential for further investigation and promises to contribute to new ways for traffic detection and management. Thus, our future work will include the following:

- Improving the Bayesian Network for data fusion including further ideas for obtaining a better and more accurate speed estimation in case of Bluetooth and an extensively analysed C2X scenario. This includes
 - o the extension of the causal dependencies of the Bluetooth likelihood node
 - o the consideration of steady-state conditions for traffic in case of specific free flow and congested flow conditions, the application of different time windows for speed estimation as a prerequisite to a certain Bluetooth detection
 - o the modelling and quantification of different C2X conditions, e.g. multi-path effects
 - o modelling and consideration of the traffic light control to speed estimation
- Application of the method on other simulation scenarios with more realistic Bluetooth equipment rates and for a longer period of time, e.g. 1 day instead of 1 hour only
- Investigation of the traffic state and velocity dependency of the correction factor for vehicle count estimation
- Take into account other sensor technologies with higher detection ranges, but also rare penetration ratios, e.g. WiFi
- Model the sensor CPDs taking into account the affection of the sensors by external (weather, illumination, multipath effects) and internal (wear and tear, signal transmission) influences

3 Further Decentralized Surveillance Applications

3.1 Driving Anomalies and Hazard Detection

Apart from expected behaviour of vehicle drivers there might occur intended or unintended deviations from these patterns, herein called driving anomalies or atypical behaviour. They could lead to hazardous situations and in the worst case to road accidents. From the viewpoint of traffic safety a detection of such driving anomalies on a disaggregated level is desired. Amongst others, the Surrogate Safety Assessment Model (SSAM) is widely adopted. However, related to the COLOMBO assumption of low equipment rates, the task is pretty much reduced to traffic incident detection and to the safety measures based on trajectories.

Incidents, either caused by traffic participants or by unscheduled external influences¹, can also worsen the general traffic conditions such as capacity reduction with travel time extensions and traffic jams, and/or lead to secondary driving anomalies. Hence traffic managers are interested in detecting such transport system shortages. To do so, Automatic Incident Detection (AID) systems and Incident Management Decision Support System (IMDSS) are offered by the industry.

Incidents comprise:

- Stopped Vehicles
- Low Speed Vehicles / Speeding Vehicles
- Traffic Congestion up to total road blocks
- Frequent Lane Change
- Opposite Direction / Shoulder Road Traveling

Incidents influence the flow of the road users involved in the incident as well as vehicles passing by. Therefore it is a reasonable approach detecting atypical behaviour as an indicator for a road incident. The incident detection system developed within COLOMBO identifies vehicle trajectories that are the result of atypical driving of road users. A traffic situation can be labelled as a traffic anomaly, if the number of road users driving in an unusual mode increases above a predefined threshold. The following questions have to be answered in this context:

1. What is a proper definition of usual and unusual (typical/atypical) driving behaviour?
2. What is the right threshold for dividing a set of trajectories into typical and atypical ones?
3. Is it possible to detect incidents and abnormal driving behaviour with a given method from V2I data at low penetration rates?
4. What incidents are detectable at what precision and reliability?

Rather than finding a general answer for all of these questions, within COLOMBO, specific assumptions were made and an incident detection algorithm and prototype were implemented and investigated. In contrast to the majority of work within COLOMBO that is based on traffic simulation, the work presented in this chapter uses trajectory data of real life road users. The following research questions were addressed within this study:

1. How well can be incidents detected in Car2X data?
2. What is the influence of the penetration rate of Car2X on the quality of incident detection?
3. What is the influence of position and speed errors in Car2X data on the quality of incident detection?

This chapter is structured as follows: First the state of the art is reviewed in section 3.1.1. The proposed incident detection algorithm is introduced in section 3.1.2. The experiments and evaluation considerations are explained in section 3.1.3. It is particularly important to model the

¹ To be distinguished from planned events like construction sites or road closures.

influence of distortion of the GPS signal on the trajectory data. Therefore the GPS noise was modelled as outlined in section 3.1.3.1. Experimental results are shown in section 3.1.4. Finally, conclusions are drawn in section 0

3.1.1 Identifying Atypical Behaviour: Review of the State of the Art

In literature, different approaches are known to distinguish between common and uncommon behaviour of moving entities in a scene. Some approaches with the representation of interactions and group activities within a scene were reported (see [21] and [22]). Most focus on analysing single trajectories, thus neglecting the interaction between road users. Because V2X data at lower penetration rates than 40% gives low information on the interaction of road users, the work within this study follows this paradigm.

A trajectory in its original physical meaning is a time-space-relation of a moving vehicle, pedestrian or bicycle. In its extended traffic engineering context it can also be a time series of further relevant object movement descriptors like velocity, acceleration, or heading angle. Thus, pattern recognition techniques can be used to identify if a trajectory represents typical or atypical motion. A well adopted pattern recognition method is the Support Vector Machine (SVM) [18]. Numerous attempts for using SVMs for classifying the behaviour of moving entities in a scene into typical and atypical ones have been reported in literature ([19] and [20]).

Since a trajectory can be regarded as a signal that exists in a multidimensional space, a dimension reduction is reasonable. In this context, the Principle Component Analysis (PCA) is a widely applied tool. The PCA maps the signal into a lower-dimensional space spanned by some orthogonal vectors, called principal components. The PCA allows representing the signal by an approximated version of itself which is a weighted sum of principal component vectors. This can be seen as a compression operation. Common or uncommon behaviour or behavioural attributes like “Walking” can be linked to a region within the principal component vector space (see [15] and [16]).

Another way for classifying trajectories is by using an Artificial Neural Networks. A respective approach is described in [23], however without providing experimental results, and in [24] using two competitive neural network topologies.

Self-Organizing Maps (SOM), introduced in [25], (also Kohonen maps) are two layer neural networks capable of learning the probability distribution of input data by vector quantization. SOMs have been applied for the problem of driving anomaly detection in [14]. Experimental results reported in [14], however, indicate that the system’s sensitivity is unsatisfying. A ghost driver, which is a very unusual driving anomaly, is assigned a score of 0.89 while the average driver has a score of 0.98. The proper choice of the threshold for separating a set of trajectories into typical and atypical ones remains a challenge.

Another widely used approach (see [17], [26], [27], [28], [29], [30], [31] and [32]) to the detection of abnormal behavior in trajectories is modeling their spatio-temporal dynamics probabilistically as activities using hidden Markov models (HMM). In the following a closer look is taken on two of the HMM approaches.

In [33], the observed vehicle trajectories were transformed to sequences of vehicle states, which are used in model estimation. The vehicle behavior was represented by a set of observable states of the HMM. Three states were used for describing the vehicle’s velocity: {acc} for a positive change of velocity, {crs} for constant velocity, and {dec} for a velocity decline within a time step. Also a set of states was used to describe the traffic condition for each vehicle, and if it was in conflict or not. After estimating the model parameters, each sequence of movements could be assigned a probability to be generated by the HMM. It was shown, that vehicles in conflict with other vehicles are less likely to be generated with the estimated HMM.

A similar approach with a stronger focus on detecting abnormal trajectories is presented in [34]. Trajectories are modeled as paths, which are represented in a compact way by activities

incorporating location, dynamics and temporal properties. Each activity is represented by a HMM. Since only normal trajectories were used for learning the activities, abnormal trajectories are expected to have a low class assignment. A decision threshold was derived during the learning phase by comparing the average likelihoods of samples in the training set and those outside. The results presented in [34] show that U-turns and other abnormal behavior can be detected successfully.

In order to decide on the method to favour for the investigations within COLOMBO, the following aspects were considered:

- Users like road operators, traffic engineers, and traffic scientists wish to understand what is going on in the system.
- Users will appreciate a visualization of the knowledge that the system uses to classify and to label trajectories.
- Users wish to control the system's threshold when it decides if a trajectory represents atypical behaviour or not. A meaningful score like 0=atypical and 1=typical would help to do so.

Reviewing the approaches described in literature, the following conclusion was drawn: Almost every of the sophisticated approaches like Neural Networks including SOM, the PCA, the HMM and the SVM give good results in detecting abnormal trajectories. It seems that the task of detecting atypical movements on the road is not very hard to solve. However, none of the more sophisticated approaches provides a human understandable representation of what is typical or atypical. Visualization can neither be easily found in literature nor deduced straightforward by comprehending the concept. Therefore, many of the concepts can be regarded as more complex than needed and on the other hand not well matching with the need of practitioners.

Therefore, a less sophisticated, more hands-on and very transparent approach for driving anomaly detection is chosen within the scope of this investigation.

3.1.2 The Probability Density Map (PDMap) Approach

There are numerous parameters that can be used to assess if the behavior of a road user is as usual or not. Reasonable parameters in this regard are location-based vehicle count, vehicle velocity, the heading angle of the vehicle and many more. The idea of a PDMap is to link the information on the expected value of these parameters to the location within the scene [14]. A given trajectory can then be assessed with respect of its likeliness to appear in a road scene as follows:

1. Examine each location (x, y) of the trajectory,
2. determine, how likely it is that a road user is present at the location (x, y) ,
3. estimate the likelihood of a road user's current velocity v of the road user at (x, y)
4. and the road users current heading angle φ at (x, y)
5. Calculate a cumulated likelihood measure over all trajectory locations

The likelihoods can be estimated using a histogramming technique. Within the investigations within COLOMBO the following parameters were used for histogramming:

- 1) The locations within the scene are discretized into bins.
 - a) The bins are located on a grid.²
 - b) For each bin, an additional two-dimensional histogram over the heading angle and the absolute velocity is accumulated.³
- 2) The trajectories are mapped onto the discretized grid by
 - a) dividing the trajectory into detection segments⁴

² Mesh widths in the range of 10...30cm work well. The mesh width of the grid used within COLOMBO is 17 cm.

³ The discretization intervals used in this investigation are 0.5 m/s for the velocity and 10° for the heading angle.

b) and determining the bins affected using a straight line drawing algorithm.

Hotspot bins where the average traffic volume is high accumulate a high number of votes while locations less frequently visited by road users accumulate fewer votes. The histogram bins are filled during a learning phase. The learning phase can be carried out once in advance. It is as well possible to adapt the histogram using new trajectory samples during online operation of the incident detection system.

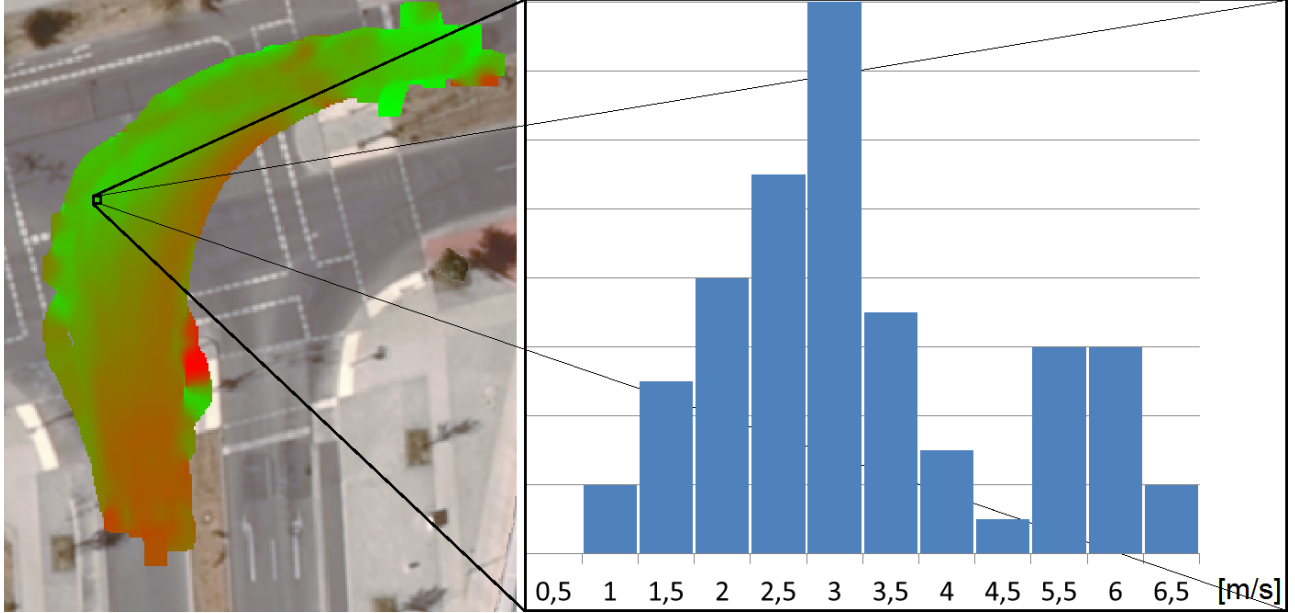


Figure 30: PDMaP principle – for each location, a histogram is recorded, reflecting the frequency of the occurrence of velocities

Equation (1) allows computing a scalar value $G_{(x,y,*)}(\vartheta)$ that converges to 1 if the trajectory $\vartheta \in \Psi$ is considered as “completely normal” driver behavior and low in the case that the behavior is “totally unusual”.

$$G_{(x,y,*)}(\vartheta) = \sqrt{\frac{1}{N_{\vartheta}} \sum_{x,y} \left(\frac{H(x,y,*)}{\max(H(x,y,*))} \right)^2} \quad (1)$$

In equation (1) the following symbols have been used:

- The quantity $H(x,y)$ denotes the accumulated number in the histogram bin associated with the position (x,y) and one of the parameters heading angle α , velocity v or count q , denoted by the wildcard symbol “*”. The quantity $\max(H(x,y,*))$ denotes highest number of votes that were accumulated in any of the bins within the positions (x,y) within the scene.
- The number N_{ϑ} denotes the number of bins covered by the trajectory as the result of the mapping process 2)b)

A combined measure for all parameters $i \in \{\alpha, v, q\}$ that characterize the trajectory ϑ can be calculated using equation (2).

⁴ A V2X CAM based trajectory contains one data record every 1.0s up to 0.1s as specified in [ETSI EN 302 637-2, 2013]. A segment is formed by two successive CAM positions which are apart between 0 and max. 4m (according to condition 1, sect. 6.3.1 “CAM generation frequency management”, ETSI EN 302 637-2, 2013).

$$G(\vartheta) = \prod_{\forall i} G_{(x,y,i)}(\vartheta) \quad (2)$$

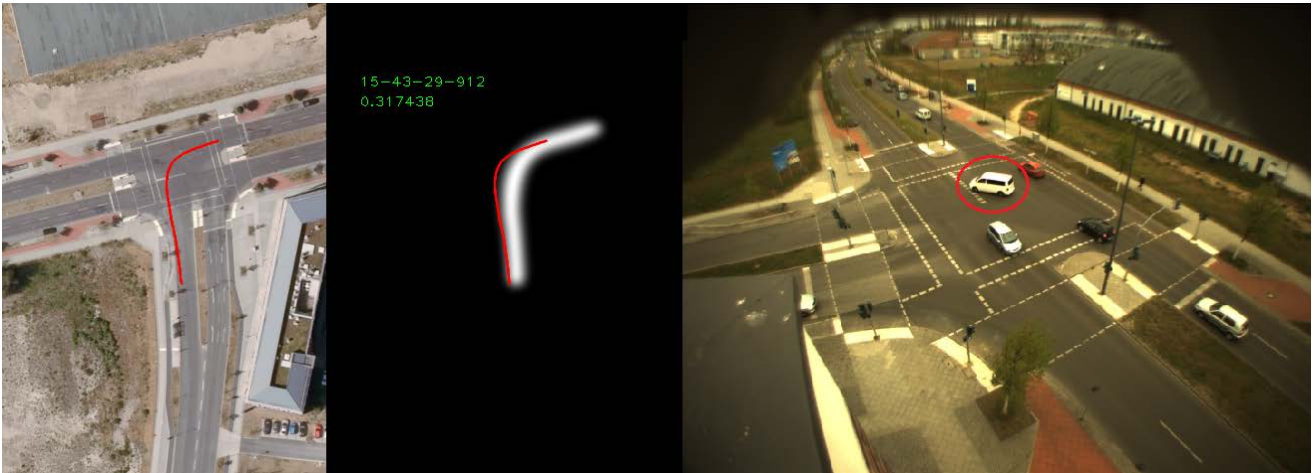


Figure 31: Atypical trajectory in a turning movement relation

An example from a road junction is shown in Figure 31. The path, that the average vehicle follows, is displayed in the center. An atypical trajectory scoring 0.31 is displayed in red. It looks like the driver is making an unusual turn here, maybe because of being distracted from road traffic. The move was correctly identified as a left turn.

Equations (1) and (2) represent the algorithm of scoring a trajectory as typical or atypical one. The following implementation details of the incident detection system are important as well:

- Road users passing an intersection may choose to pass straight or making a turn. Correspondingly, it is reasonable to use different histograms for each relation. In the learning phase, the trajectories must be first assigned to the different relations then. A proper way to do this is deciding upon the starting and ending points of the trajectory to which relation it belongs. The respective clustering operation is made using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN⁵) scan algorithm. During online operation, the scores of each incoming trajectory relating to each relation can be computed. The highest of the scores per relation give the right trajectory-relation assignment (see [14]).
- It is reasonable to use a smoothing algorithm when processing the histograms in the learning phase. Smoothing avoids black holes in the histogram. For this purpose, a truncated Gaussian mask was used for blurring the bins of the same class over neighboring positions. An 11 Pixel wide mask was used for blurring the bins of the velocity and heading angle histograms and a 25 Pixel wide mask was used for blurring the vehicle count histogram.
- A sliding window can be applied to the trajectory data if one wishes to have a short response time. Within this study, a sliding window over 20 trajectory samples was used, which corresponds to 1.6s.
- The threshold for distinguishing between typical and atypical behavior, can be set between 0 and 1. Within this study, a value of 0.1 has been applied.

Compared to many other approaches for determining atypical situations, the PDMap approach has the following advantages:

- The maps can be plotted and give a meaningful representation and visualization of the average road user behavior. PDMap plots can be used to inform the road operator about the normal behavior of road users.

⁵ <https://en.wikipedia.org/wiki/DBSCAN>

- Trajectories with atypical properties can be displayed as an overlay over the PDMap histogram giving the operator an impression on the situation
- The scalar numbers calculated using (2) represent a sensitive score for assessing the properties of a trajectory. They are easy to interpret. The following example scores were reported in [14]:
 - a) normal track 0.7,
 - b) slower than average vehicle 0.5,
 - c) faster than average vehicle 0.25,
 - d) Ghost driver 0.0013.

Within COLOMBO, a study on incident detection using PDMaps for driving anomaly detection was conducted.

3.1.3 Experimental Setup and Investigational Approach

For the investigation, a scenario data set of trajectories of real vehicles was used. These trajectories were collected using a dedicated video surveillance system and generated by means of an advanced image processing method that allows tracking vehicles over a range of more than 100m (see [13] for details). This range is in the order of magnitude of the range of V2X communication. Therefore, the trajectory data set is considered to resemble well the real world V2X data to be expected at different V2X equipment rates in the future. Using real world vehicle trajectory data allows skipping the task of finding a scheme for generating simulation based data on incidents on the one hand and on the other hand improves the quality of the conclusions drawn from the analysis.

The block diagram of the setup (without the optional SUMO involvement) can be seen in Figure 32.

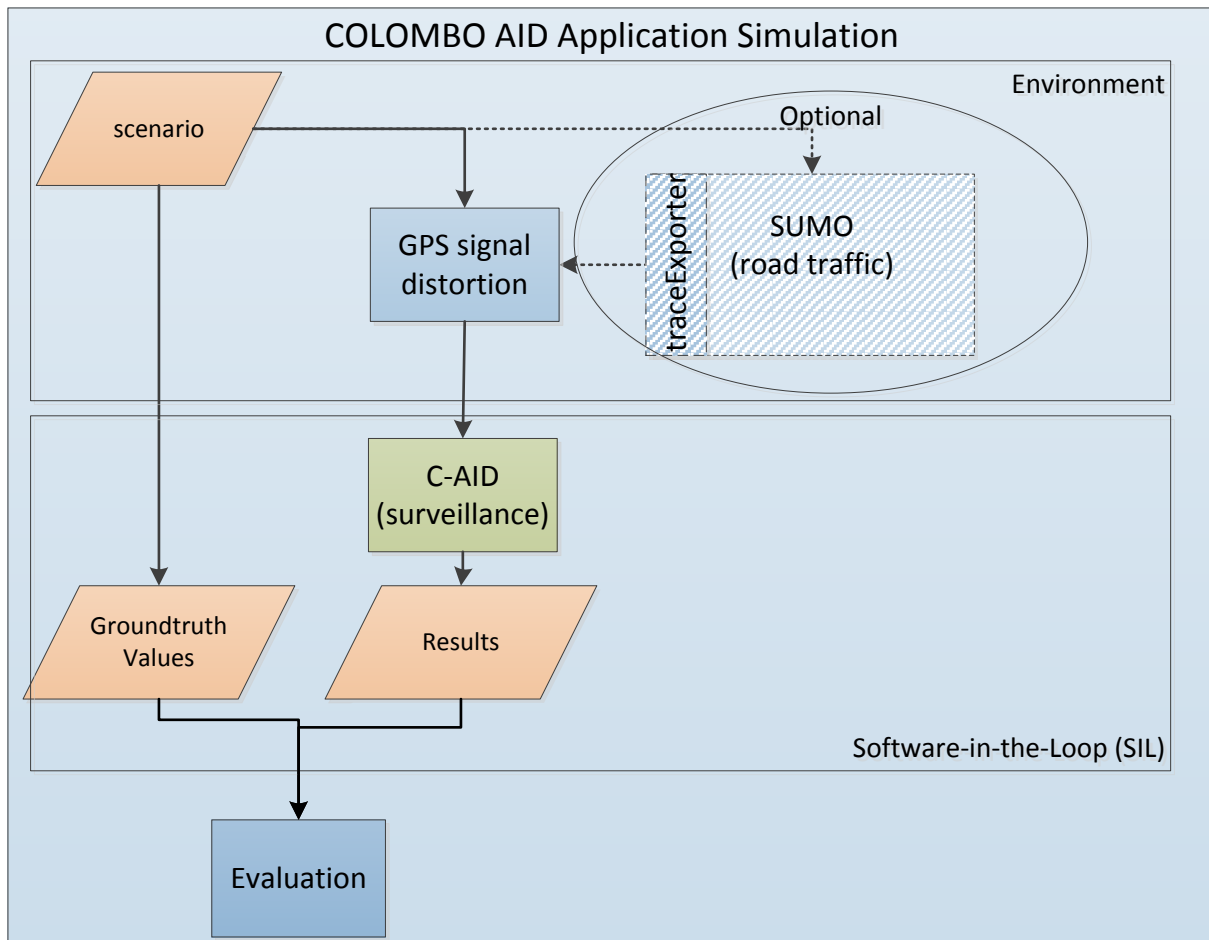


Figure 32: AID Investigation flow chart

The 3 hour video stream of a road scene was recorded at the sub-urban L625 road between Braunschweig's districts Bienrode and Waggum where it has a level crossing with a single-track railway line. The advantage of this scene regarding this study is, that road block events occur frequently for about 60s on the level crossing every time a railroad train passes (Figure 33). Recording at a frame frequency of 12.5 Hz comprises the time between 13:29 and 16:23 on Monday 1st December 2014. Within that time 1,667 vehicles on both directions were counted, the respective load curves are given for 15-min-intervals in Figure 35. Resulting average headways lay between 10s and 20s. Local speeds are on average between 8 m/s and 11 m/s, leading to trajectory lifetimes between 10s and 14s where each moving vehicle is in the detection field. As a result the average detection density on each lane is between 0.5 veh/frame and 1.4 veh/frame at any time.



Figure 33: Images of roadblock incidents taken in the road scene

Another event is the stopping of a parcel delivery car at the border of the road. The stopped car is an obstruction for a lorry and several passenger cars during the following sequence of images ("Parcel-Van"-incident Figure 34).



Figure 34: Images of the "Parcel- Van" incident

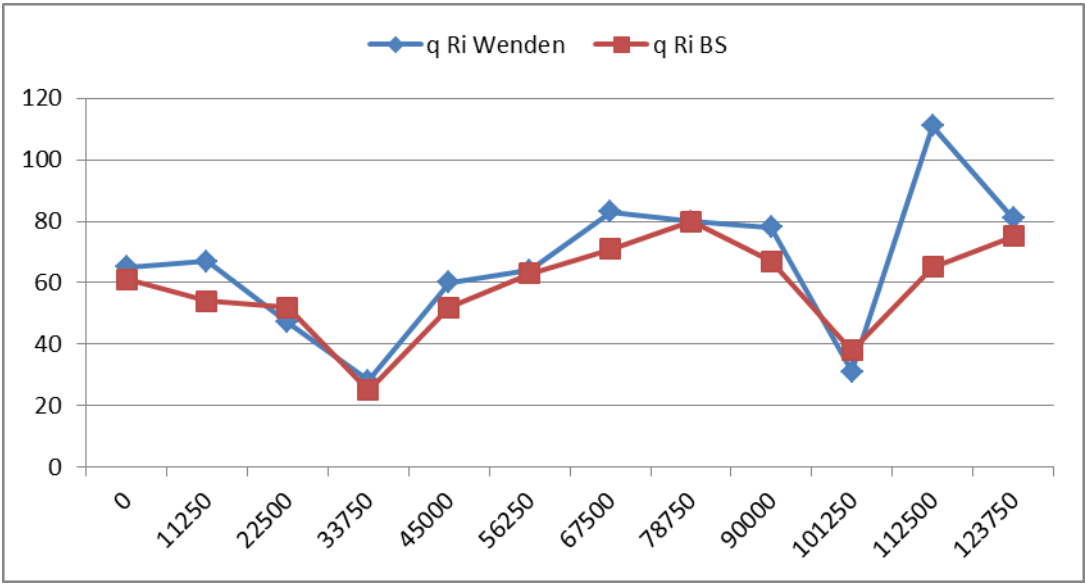


Figure 35: Traffic volumes in 15 minute intervals for the both directions “Wenden” and “BS” of the scenario

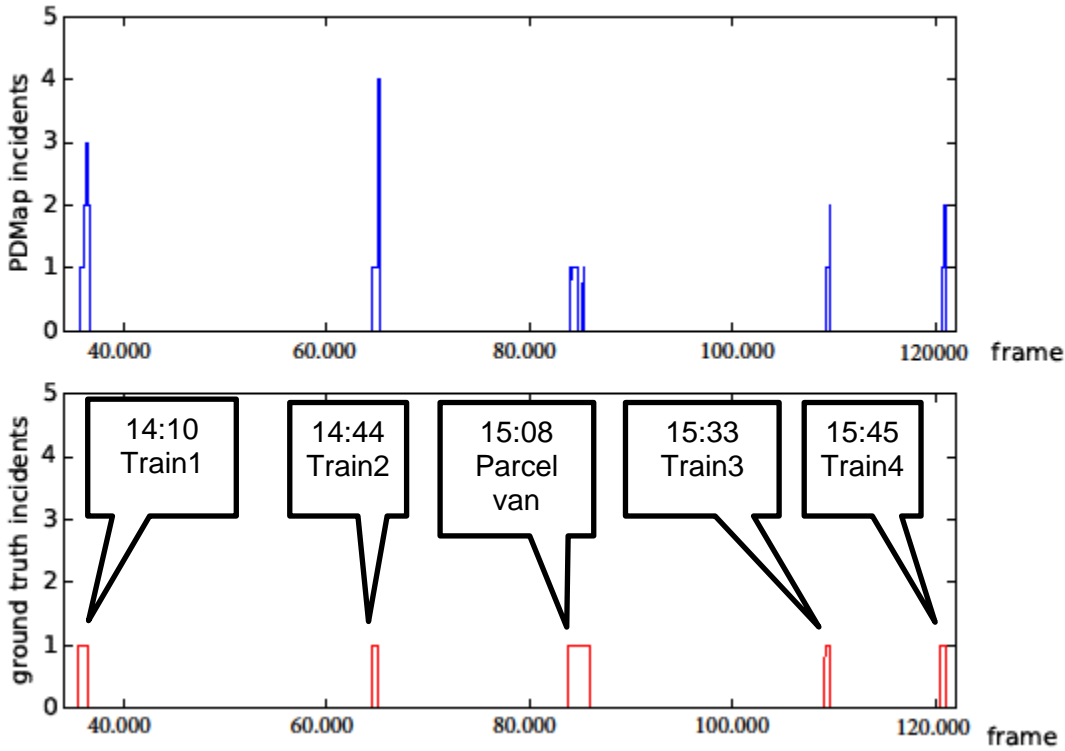


Figure 36: Upper: The number of detected atypical trajectories simultaneously detected in the frame at 100% penetration rate
Lower: Annotated ground truth for the road block incidents

For answering question 1, the scene was manually annotated⁶ in order to generate the ground truth information. Road blocks upon passing trains and the stop of the parcel van were marked as incidents (Figure 36). The quality of detection was measured by calculating the elements of a confusion matrix. The concept of the confusion matrix (see [35]) comprehends calculating how frequent the incident detection system signals

- an incident, while ground truth says there is an incident (true positive),
- an incident while there is no real incident (false positive),
- no incident while there is no real incident (true negative), and
- no incident while the ground truth says there is an incident (false negative).

The confusion matrix elements were calculated

- a) per frame and
- b) per incident.

There is some inherent bias in method a), because if the incident detection system detects the incident earlier and/or later than it is marked in the ground truth, then false positive detections will be reported, although a human observer would judge that there is no failure in the incident detection system.

This bias is not present in the evaluation method b). However, method b) cannot be implemented in a fully automated fashion. A human observer needs to judge every timeline (Figure 36) for every scenario that needs to be evaluated.

In order to deal with question 2, different C2X equipment rates $\eta = [1; 2; 5,10; 20; 50; 100]\%$ were simulated by taking out trajectories from the scenario data set. For all penetration rates $<100\%$, the tests were repeated 10 times.

Question 3 was addressed by superimposing modelled noise of the position signal using a Gauss-Markov process.

3.1.3.1 Modelling the Distortion of the GPS Position Signal

There are just a few publications referring V2I data and position accuracy in particular. According to [36], the GPS horizontal error without post processing is

- 3.26m CEP,
- 5.42m R65 and
- 22.9m R95.

Where CEP stands for “Circular error probable” and means that radius of the circle containing 50% of data is 3.26m⁷. R65 and R95 denote the error radius of circles containing 65% and 95% of measured position samples respectively. The error distribution for GPS position errors is assumed to be log-normal and can be approximated with $\mu = 1.18\text{m}$, and $\sigma = 1.4\text{m}$.

These errors are correlated. The measured positions have a certain error distribution in terms of vector length, but are not isotropically and rather slowly changing over time. Reasons for this correlation and the GPS errors in general are mainly the number and position of GPS satellites and vehicle’s velocity. In state of the art vehicles, map matching, data fusion (odometry) and other algorithms are used to improve the precision.

Within this study, a best case and a worst case scenario were considered: In the best case scenario Kalman-filtered raw trajectories generated by image processing were used. The data quality to be

⁶ Annotation in this context means that frames where the incident was visible in the video was marked.

⁷ https://en.wikipedia.org/wiki/Circular_error_probable

expected from V2X GPS-trajectories with state-of-the-art post processing is assumed to be reflected well by this data. In the worst case scenario raw trajectories were first distorted with a Gauss-Markow process and Kalman-filtered afterwards (for filter parameters see Appendix A).



**Figure 37: Top: original trajectory after Kalman-Filtering (Best Case).
Middle: trajectory after adding noise generated by Gauss-Markow process.
Bottom: trajectory after Kalman-Filtering the noisy trajectory (Worst case)**

According to [37], the correlation of the GPS position errors over time can be modeled by a Gauss-Markov process and simulated using equation (3).

$$x_{k+1} = e^{-\beta\Delta T} x_k + w_k \quad (3)$$

In equation (3), w_k denotes random samples of a zero-mean Gaussian distribution. Within this study, the following assumptions were made based on the information given in [37]:

- The GPS receiver uses carrier phase smoothing.
- Therefore the variation of the Gaussian noise component is $\sigma_g = 1.95m$
- For the same reason, the Gauss-Markov process timing constant was set to $= \frac{1}{1100s}$.
- The measurement noise was neglected because of its very low amplitude of $0.0019m$.

Further, it was assumed that the GPS-receiver independently measures a speed signal via the Doppler Effect. The error of this signal was not simulated for two reasons: firstly, no appropriate error model could be found in literature, and secondly, the relevance of the error was estimated low, because speed measurement errors can very well be compensated by data fusion with the odometer signal of the vehicle.

3.1.4 Experimental Results

The images in Figure 38 to Figure 40 contain visualizations of the PDMap histograms of the different quantities introduced in section 3.1.2 (velocity v and count q , the PDMap of the heading angle α is omitted). These visualizations were produced by the graphical user interface of the automated incident detection system. Especially the local vehicle count view in Figure 39 gives a good visual impression to the operator about the expected normal behavior of the road users.

Notes:

- Artifacts that stem from Kalman-Filtering can be observed on the head of the green trajectory cluster in Figure 38.
- The precision of the positions decreases when the distance between vehicles and camera increases. This is due to the precision of the image processing system. Correspondingly, the tail of the green cluster is unraveled.
- The map of velocities in Figure 40 displays the average velocity calculated from the histogram of velocities.

Figure 35 upper part shows the number of atypical trajectories detected in a frame. One can see that this number does not exceed five vehicles, which is due to the detection range of the camera as well as due to the fact that the road block incidents often last for a short time.

The task of distinguishing between typical and atypical trajectories becomes more challenging when the position noise of the GPS is added (see Figure 41). However, the confusion matrix values in Table 3.1 to Table 3.4 show that there is almost no influence of the noisy GPS measurements on the incident detection accuracy. This gives a positive prospect on the robustness of the PDMap approach.

The detection quality scales with the V2X penetration rate as can be seen in Table 3.2.

As a rule of thumb, the true positive detection rates per frame (Section 3.1.3 method a) seem to be about four times lower than the true positive rates measured per incident (Section 3.1.3 method b). This can be attributed to the fact that the number of vehicles involved in the incident influences the detection rate of this incident. This number is about four in this scene⁸. Therefore one may expect, that the detection rate for an incident is related to both the equipment rate and the number of cars involved in that incident. Large traffic jams will be easier to detect at low C2X penetration rates than small incidents.

⁸ Figure 5 displays the number of detected cars at the same time. However, the number of cars involved in the incident can be higher when cars get into the incident sequentially.



Figure 38: Trajectory assignment after clustering (Green: Cluster 1, Red: Cluster 2)



Figure 39: PDMap of the vehicle count (Red: relative frequency is high, Green: relative frequency is low)



Figure 40: PDMap of the vehicle velocity (Red: $v=17.3\text{m/s}$, Green: $v=0$)

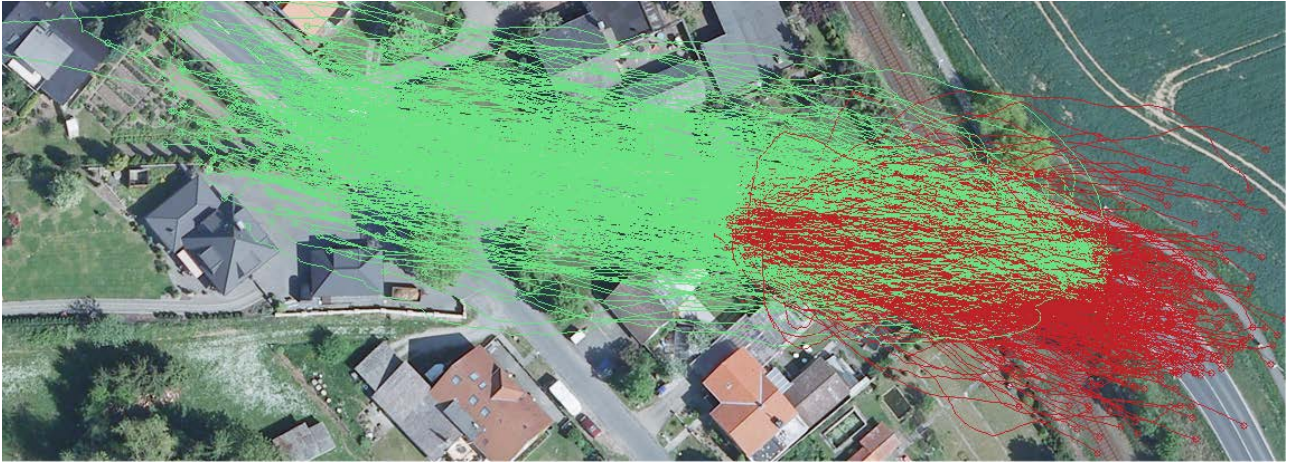


Figure 41: Trajectory assignment after clustering for the worst case trajectory data set (Green: Cluster 1, Red: Cluster 2)

Table 3.1: Confusion matrix values for different penetration rates and undistorted trajectories calculated per frame (Section 3.1.3 method a)

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
TP	41.0%	23.1%	10.8%	4.8%	3.9%	0.2%	0.6%
FP	5.5%	4.0%	2.2%	0.8%	0.7%	0.2%	0.1%
TN	59.7%	59.7%	59.8%	59.8%	59.8%	59.9%	59.9%
FN	0.8%	1.4%	1.8%	2.0%	2.0%	2.2%	2.2%

Table 3.2: Confusion matrix values for different penetration rates and GPS noise added calculated per frame (Section 3.1.3 method a)

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
TP	41.0%	23.0%	11.7%	4.8%	1.3%	2.5%	0.0%
FP	5.6%	3.6%	2.4%	1.4%	0.4%	0.5%	0.0%
TN	59.3%	59.3%	59.4%	59.4%	59.4%	59.4%	59.5%
FN	0.8%	1.4%	1.8%	2.0%	2.1%	2.1%	2.2%

Table 3.3: Detection rates for incidents at different V2X equipment rates for undistorted trajectories calculated per incident (Section 3.1.3 method b)

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
TP	100%	84%	48%	22%	14%	4%	4%
FP	0%	0%	0%	0%	0%	0%	0%
TN	100%	100%	100%	100%	100%	100%	100%
FN	0%	0%	0%	0%	0%	0%	0%

Table 3.4: Detection rates for incidents at different V2X equipment rates for trajectories with GPS noise added calculated per incident (Section 3.1.3 method a)

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
TP	100%	80%	48%	26%	8%	10%	0%
FP	0%	0%	0%	0%	0%	0%	0%
TN	100%	100%	100%	100%	100%	100%	100%
FN	0%	0%	0%	0%	0%	0%	0%

3.1.5 Discussion and Future Prospects

An incident detection system has been designed, implemented and tested within COLOMBO. The system relies on trajectories that can be recorded from V2X data. Tests were made in a real-world scenario over a time interval of 2:54:22. The scenario contains five incidents – four road blocks and one traffic flow obstruction. The system proves to work well. Within the scenario, it did not produce any false alarm and reliably detected every incident for a simulated V2X penetration rate of 100%. The system exhibited notable robustness against noise added for simulating the position distortion of the GPS signal.

Tests for different penetration rates, as expected, show an almost linear decline of the true positive detection rate at lower penetration rates. It is hypothesized that the proportionality factor is the number of vehicles on average affected by the incident. The conclusion is that a large traffic jams and long lasting events can be detected well at low V2X penetration rates, while small events are less likely to be monitored. This corresponds to the needs of a traffic light controller or a road operator. Future work will concentrate on fusing the information about single atypical trajectories in order to determine the type of incident.

3.2 Local Emissions Monitoring

Transportation takes a major role in the production of pollutant and noise emissions [1]. In order to reduce vehicle emissions it is necessary to measure or estimate the real emissions [2]. The most common method to monitor pollutant emissions from road traffic is air quality measurement performed at road side locations. This allows for an assessment of the contribution of vehicle emissions when the measured air concentrations are compared with concentrations from a

representative background measurement (e.g. for a city area: a close to town off-traffic site). This method is affected with several uncertainties, as

- For several important pollutants (e.g. PM₁₀, CO₂), the background concentrations are close to the roadside values
- Air quality measurement is always affected by meteorological boundary conditions (wind conditions, precipitation etc.)

Another method to investigate vehicle emissions in on-road conditions is to use portable emission measurement systems (PEMS) which are directly attached to the tail pipe [3]. Vehicles equipped with PEMS have to pass a certain road section several times in each scenario (e.g. in each traffic light control variation or each speed limit) and the measured emissions can then be compared. However, the use of PEMS is costly and in general only few systems are available (usually a single system is owned by a lab). So only a small sample of vehicles can be monitored, which results in uncertainties in the representativeness of the recorded driving behaviour and even more of the recorded emission results. The vehicle fleet consists of various different propulsion concepts (e.g. gasoline or diesel engines, different exhaust gas after treatment systems) which show different sensitivities of emission behaviour to particular driving conditions.

Measuring emission in a real test field is time consuming and difficult to set up. Therefore there are other research studies which focus on other ways how emission can be estimated. The research study of [4] compared emission estimations with the measured air quality. The authors state that a detailed knowledge of the vehicle fleet and the speed profiles is needed to properly estimate the local emissions. In [5] an approach for traffic monitoring and the computation of vehicle emissions by Floating Car Data (FCD) is presented. FCD are usually taxis (or other vehicles) which are equipped with an FCD device to send the current GPS (Global Positioning System) position. From FCD the current travel times for the taxi routes can be estimated. In this study the travel time is used to estimate the current traffic state to compute the estimated emissions.

The advantage of Vehicle2Vehicle or Vehicle2Infrastructure (V2X) communication is that a speed profile for every equipped vehicle is generated that can be directly used for the estimation of the emissions. The problem is only that it will take a long time until all vehicles will be equipped with V2X communication. To overcome this problem this paper presents an approach which combines the detailed speed information from V2X communication with the data from induction loops to estimate the local vehicle emissions.

3.2.1 Simulation Scenario

The local emission monitoring approach was tested using the traffic simulation SUMO [6]. For the European Commission project COLOMBO an extension to SUMO for emission simulation was made which is called PHEMlight [7]. PHEMlight is a simplification of the model PHEM (Passenger Car and Heavy Duty Emission Model) [8] and provides a simple but accurate simulation technique to derive modal emission quantities for given speed and acceleration demands. To avoid high storage demand and long computation times the history driving trajectories (speed and acceleration) transient dynamic corrections, temperature influences on after-treatment-systems and the driver gear shift model to compute engine speeds were not included in the simulation. These functions from PHEM were replaced by generic functions for a single car which describes fleet average values. In combination with SUMO, PHEMlight uses the vehicle speed and acceleration data from SUMO to compute the fuel consumption and emission for each vehicle.

The simulation evaluation was performed with different simulation scenarios which are described in the following. For the investigation of the different approaches a simulation scenario of a single

four legged intersection was simulated (see Figure 42). In each direction the same static volume of vehicles drive straight forward, turn left and turn right (each 2,000 vehicle). Each monitoring approach was simulated 10 times spanning 7 hours. The emissions were calculated every 5 minutes. The simulation was run with different penetration rates of equipped vehicles: 1%, 10%, 20%, and 50%.

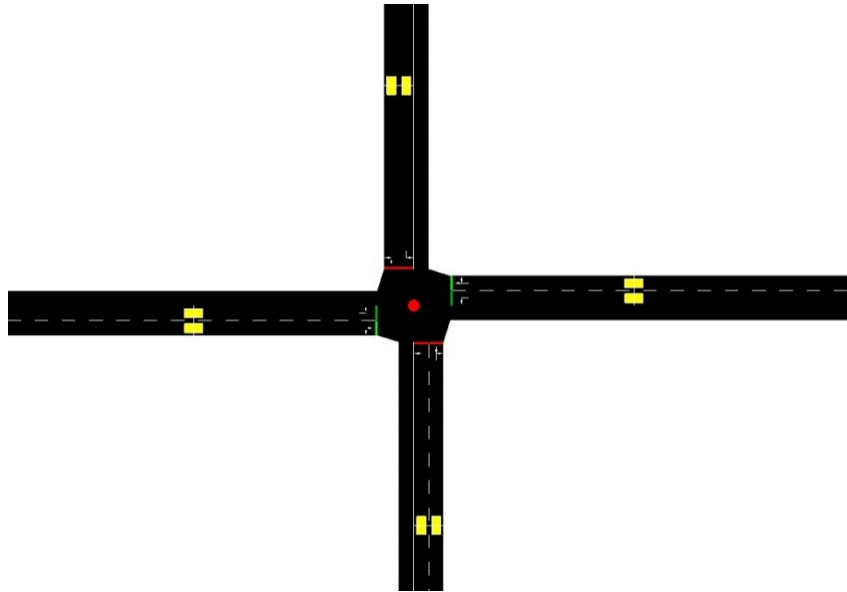


Figure 42: Basic intersection in SUMO (yellow: induction loops, red: Road Side Unit)

3.2.2 Emission Monitoring Approach

The basic idea for emission monitoring is to use the V2X data of equipped vehicles and additionally the measurements from conventional traffic detectors (see Figure 43). In the early adoption phase of V2X communication there will probably be only a low number of vehicles with V2X communication equipment. Therefore, it will be difficult to estimate the total emissions of an intersection by taking the data from V2X communication only into account. Additionally, the count and optionally vehicle classification data from inductive loops can help to get a picture of the whole traffic situation. While inductive loops are widely used for calculating traffic flows and adapting traffic light phases, inductive loops alone cannot be used for emissions estimations (because they only measure the number of vehicles at one location). For the emission calculation only one intersection will be monitored. For further research the basic approach can be extended to monitor larger regions as well.

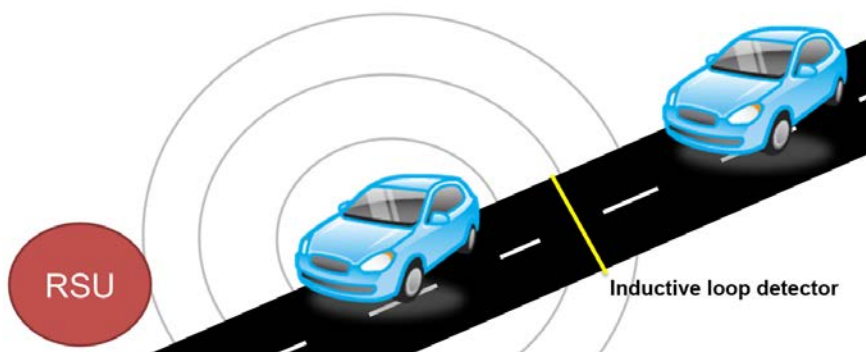


Figure 43: Data collection via a road side unit and an inductive loop detector

The Emission Monitoring System (EMS) was implemented with an application which uses SUMO for the traffic simulation and the PHEMlight for the Emission simulation. Alternatively, the produced emissions could be calculated via the traceExporter within SUMO and PHEM. The

ground truth for the emissions is estimated via the calculated emissions of all vehicles within the simulation scenario. The EMS application uses only the trajectory data of the vehicles from the simulation for the equipped vehicles only. Afterwards PHEM/PHEMlight is used to calculate the emissions from the trajectory data. The calculated results are compared to the ground truth in the evaluation process (see Figure 44).

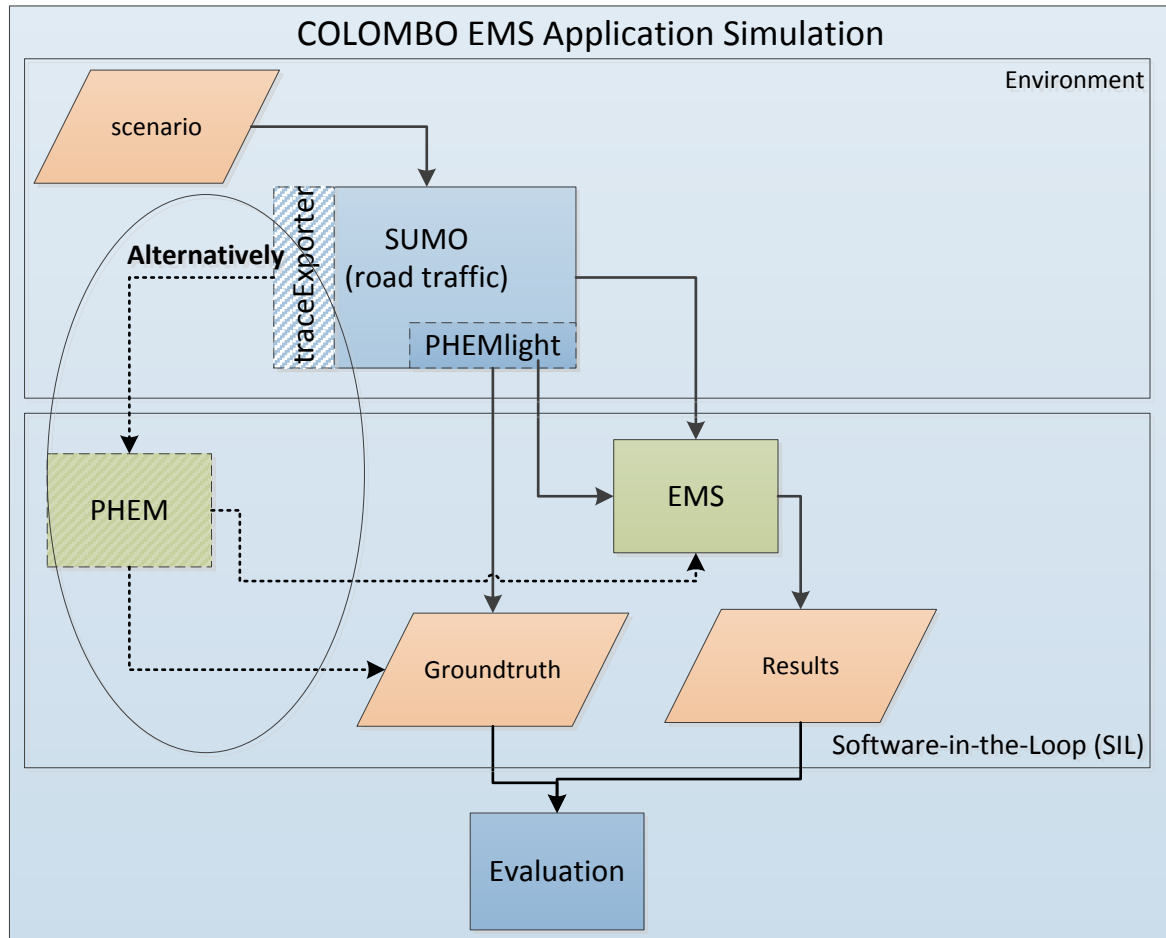


Figure 44: Implementation of the Emission Monitoring System with the COLOMBO System

3.2.2.1 Assumptions

There are some general assumptions for all analysed approaches which are described here:

- Inductive loops: every incoming lane of the intersections has its own inductive loop detector which determines the vehicle counts and the vehicle types, but not necessarily the speed.
- Intersections: At each monitored intersection the existing road side unit (RSU) collects the V2X data. For the investigation (the antenna of) the road side unit is placed in the middle of the intersection to provide a good reception range within all approaching lanes of the intersection.
- CAM definition: equipped vehicles are broadcasting so-called “Cooperative Awareness Messages” (CAMs) as defined by ETSI [9]. The following data fields from CAMs are used for this research:
 - station characteristics (in this case private vehicle)
 - vehicle type
 - vehicle Speed, vehicle speed confidence
 - curvature, curvature confidence
 - signed acceleration in direction of the node heading
 - position confidence ellipse

- V2X vehicle positioning (GPS) is not accurate enough to determine when it is passing the loop or on which lane the vehicle is driving on.
- The emission monitoring should also work with low penetration ranges.

3.2.2.2 Basic Approach

The simplest approach analyzed here is called “Basic Approach” in the following. The CAMs of all equipped vehicles are collected in the RSU. From the CAM data fields speed a time-speed series can be computed. The emissions of the equipped vehicle can be calculated by feeding this time-speed series into the emission model PHEMlight which runs as an application inside the RSU. The total number of vehicles which have passed an inductive loop detector at this intersection within a certain time period is counted. From these data the total number of vehicles, the percentage of equipped vehicles, and the emissions of the equipped vehicles are known. Hence, the estimated emissions for all vehicles can be calculated for this:

$$E_{total} = \frac{E_{V2X}}{q_{V2X}} \times q_{total}$$

Where:

E_{total} : estimated total emissions

E_{V2X} : Emissions of equipped vehicles

q_{V2X} : Equipped vehicles

q_{total} : total vehicles

This approach can be refined by distinguishing between the different vehicle types/classes which can be detected by the loops.

$$E_{total} = \sum_{vtype} \frac{E_{V2X}^{vtype}}{q_{V2X}^{vtype}} \times q_{total}^{vtype}$$

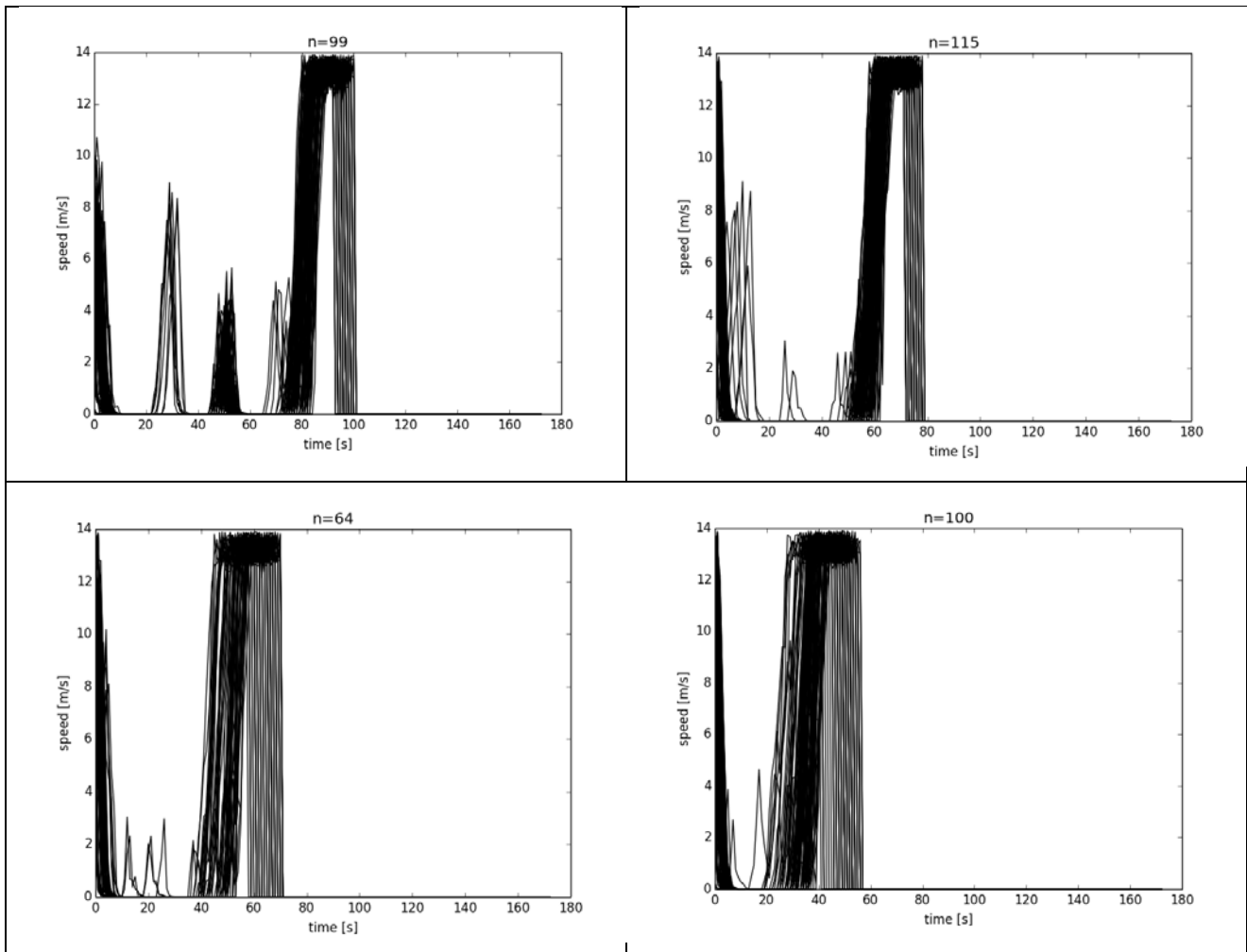
This basic approach is very simple and does not take into consideration the differentiation of driving patterns. At some intersections it might be that vehicles which have to turn left have to wait longer to pass the intersection because of opposing traffic or there could be a traffic jam at one street of the intersection and vehicles at this street produce more emissions etc.

3.2.2.3 Clustering Approach

To overcome the uncertainties of the basic approach the hypothesis is introduced that vehicles with a similar speed profile produce similar emissions. The basic approach was extended by a clustering of the speed-time series of the equipped vehicles. The road side unit stores the computed speed-time series of every equipped vehicle in the communication range. They are clustered using a Python library called “fastcluster” which provides a hierarchical clustering algorithm [10]. Wards method was used as method for the inter-cluster distance. A definition of implementation of the “ward” method can be found in [11]. One problem with the clustering algorithm was that the speed-time series had to have the same length of measured data values. Otherwise it was not possible to cluster the speed-time series. To solve this problem the maximum length of all speed-time series was determined. All speed-time series with less data points were enlarged by appending zero-values as representation that the vehicle was not driving anymore in the communication range and as a result is not producing any emissions. This way of solving the problem would be not a good solution for matching the driving profile but for emission estimations it was the simplest way.

As a first step the clustering algorithm was tested with the speed-time series of a single simulation run with a penetration rate of 1 %. Afterwards, the clustering algorithm was executed

with the speed-time series of the simulation with different sizes of clusters (from 2 to 49 clusters). As an example the different speed-time series with 6 clusters are presented in Figure 45. It can be seen that the clustering looks relatively good. It seems that a major indicator for the cluster is the actual time the vehicles spent within the communication range. Only the speed-time series from cluster number 6 are not very similar, hence it is a “junk cluster”. The idea was that speed-time series within the same cluster should have similar emissions. Therefore the emissions of the clusters were also analysed. In Figure 46 the emissions can be seen for 6 clusters. The figures show that the emissions of the speed-time series in the same cluster are also similar, only for one cluster (cluster number 6 in the example) the emissions have a large spreading. This behaviour can be seen in all analysed number of clusters. Even for 49 clusters there is one cluster which has a large spreading for the emissions. Figure 46 shows on the right side the average standard deviation of the emission clusters. It can be seen that for a small number of clusters the standard deviation is very high compared to a large number of clusters. Therefore it is recommended to use this approach with a larger number of clusters to receive good results. Furthermore, it can be seen that between 25 and 50 clusters the average standard deviation is changing only slightly. Thus, for this intersection it can be assumed that a cluster size of 25 should be detailed enough for emission evaluations.



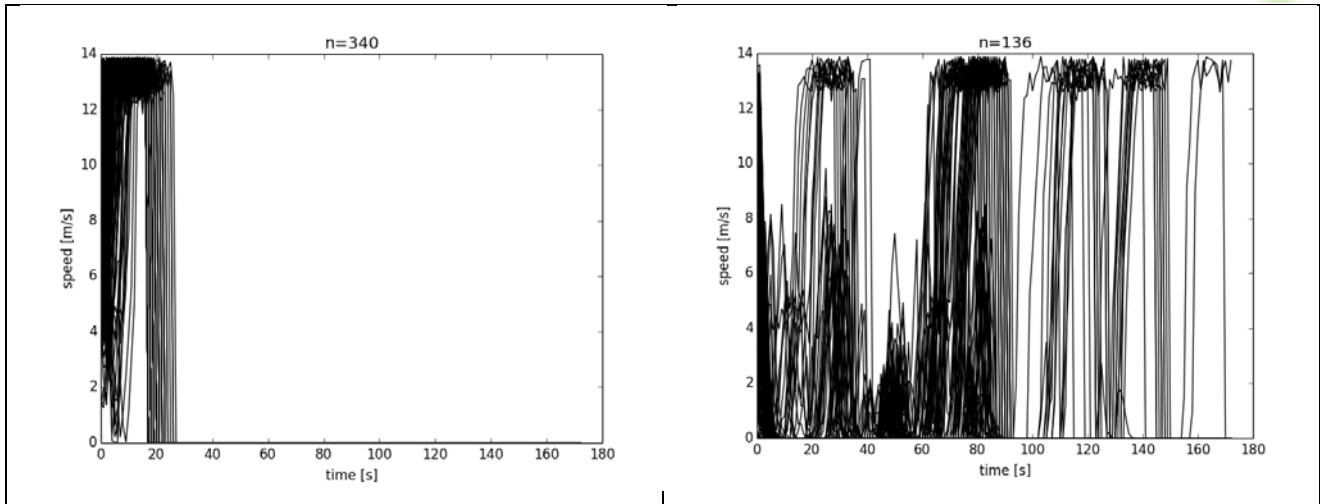


Figure 45: Trajectories with 6 clusters

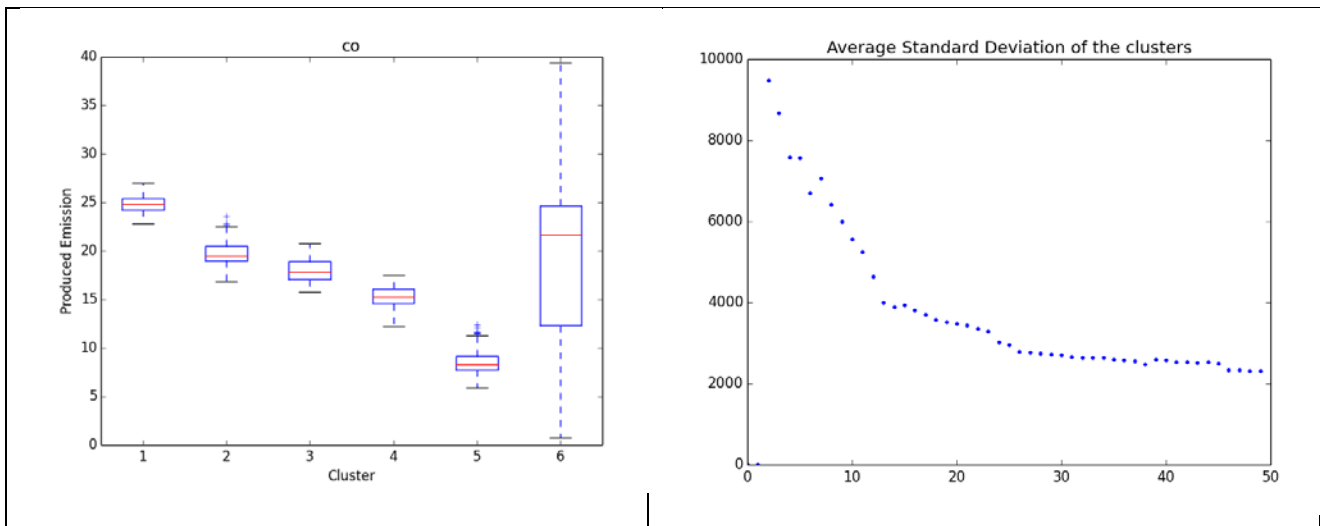
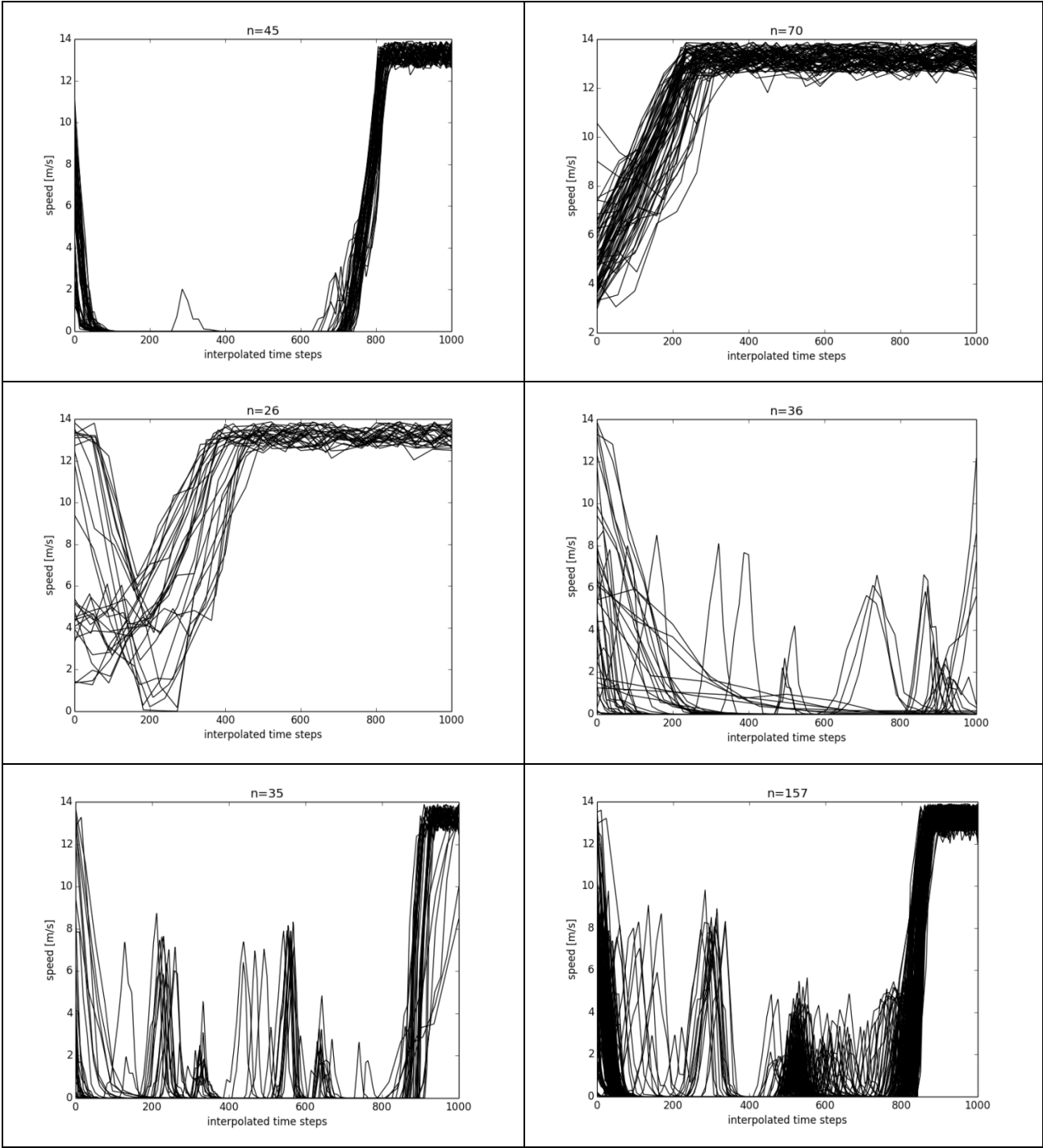


Figure 46: Relative Error of the estimated emissions over time (left) Average standard deviation of the emissions for a different number of clusters (from 2 to 49) (right).

3.2.2.4 Clustering of normalized speed-time-series

This approach was introduced to find a better solution for the problem of different speed-time-series lengths. The difference to the above described approach is that the speed-time-series are not artificially extended but are normalized over the time within the communication range. Normalization is done by interpolating between two adjacent data points. The clustering algorithm is then executed with the interpolated data points. The results of this clustering approach for an example can be seen in Figure 47. The clustered trajectories are looking also very similar within the clusters, but the emissions of the trajectories within the clusters have a much higher spread (see Figure 48).



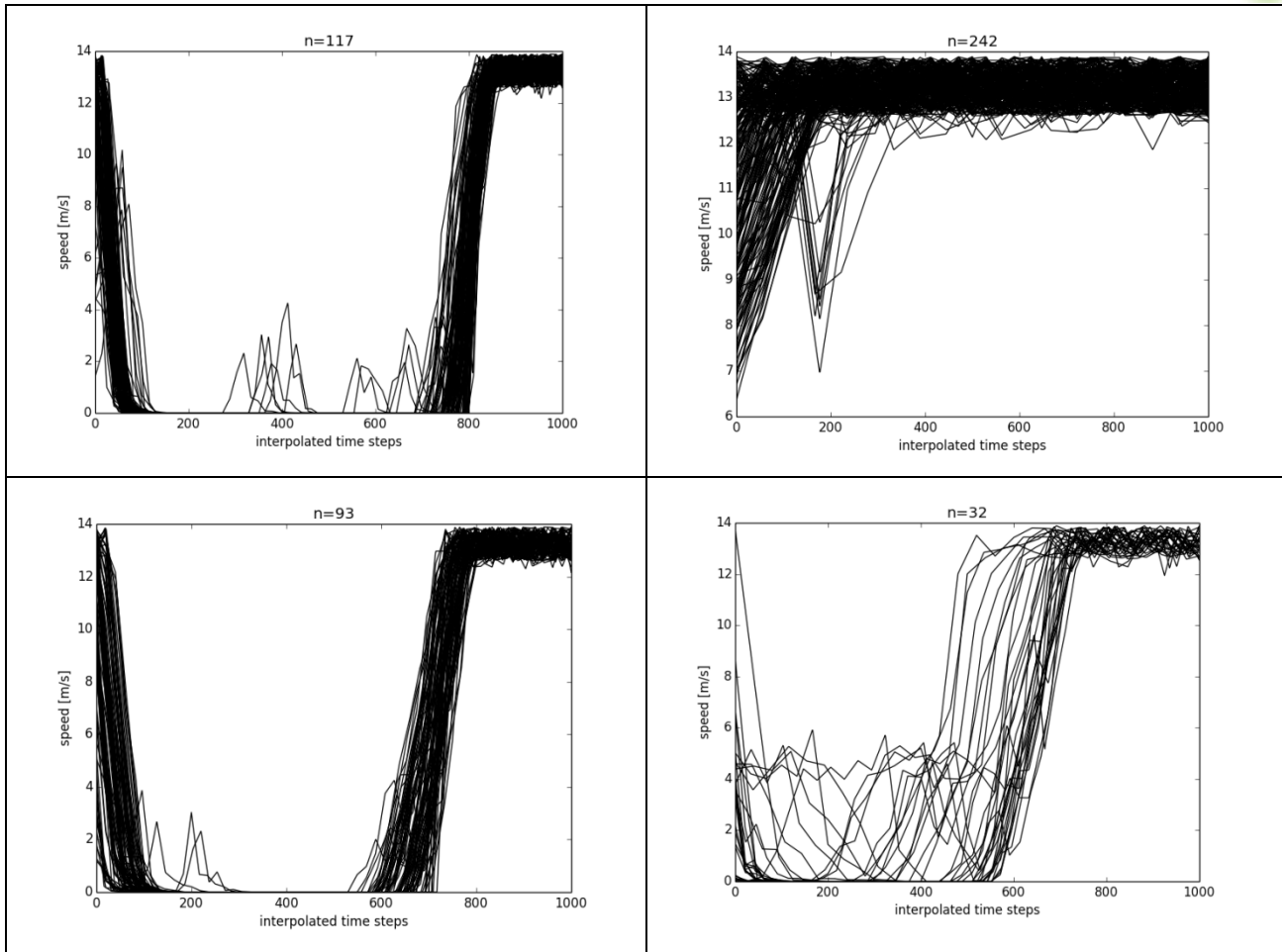


Figure 47: Results of the clustering approach with normalized speed-time-series

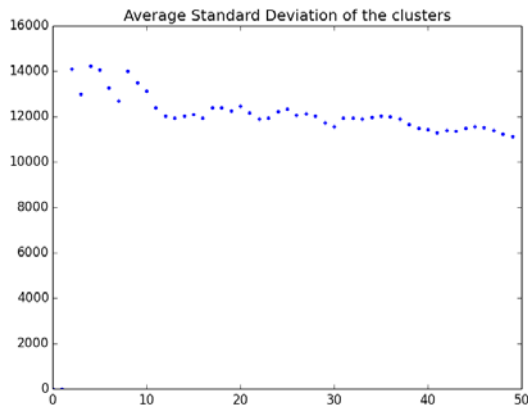
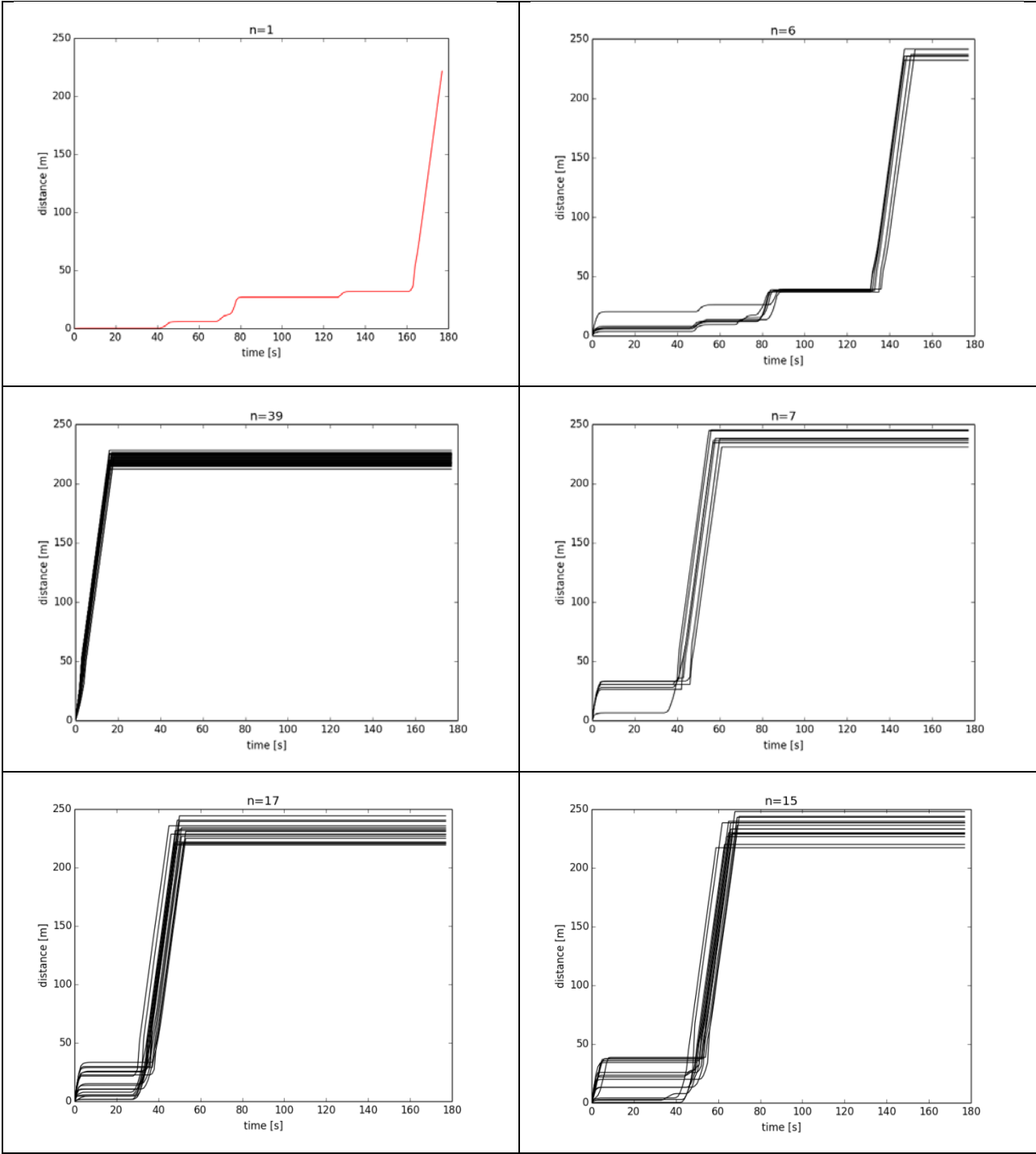


Figure 48: Average Standard Deviation of the clusters for the normalized speed-time series

3.2.2.5 Clustering of the distance over time

Another approach for clustering was to use the distance instead of the speeds of each vehicle. The results can be seen for 10 clusters in Figure 49. Some of the clusters have only very few trajectories, but the trajectories are looking similar to each other. The average standard deviation of the clusters for the emissions can be seen in Figure 50. The emission results are better for this approach than the results from the normalized speed-time series, but worse for the results of the basic clustering approach.



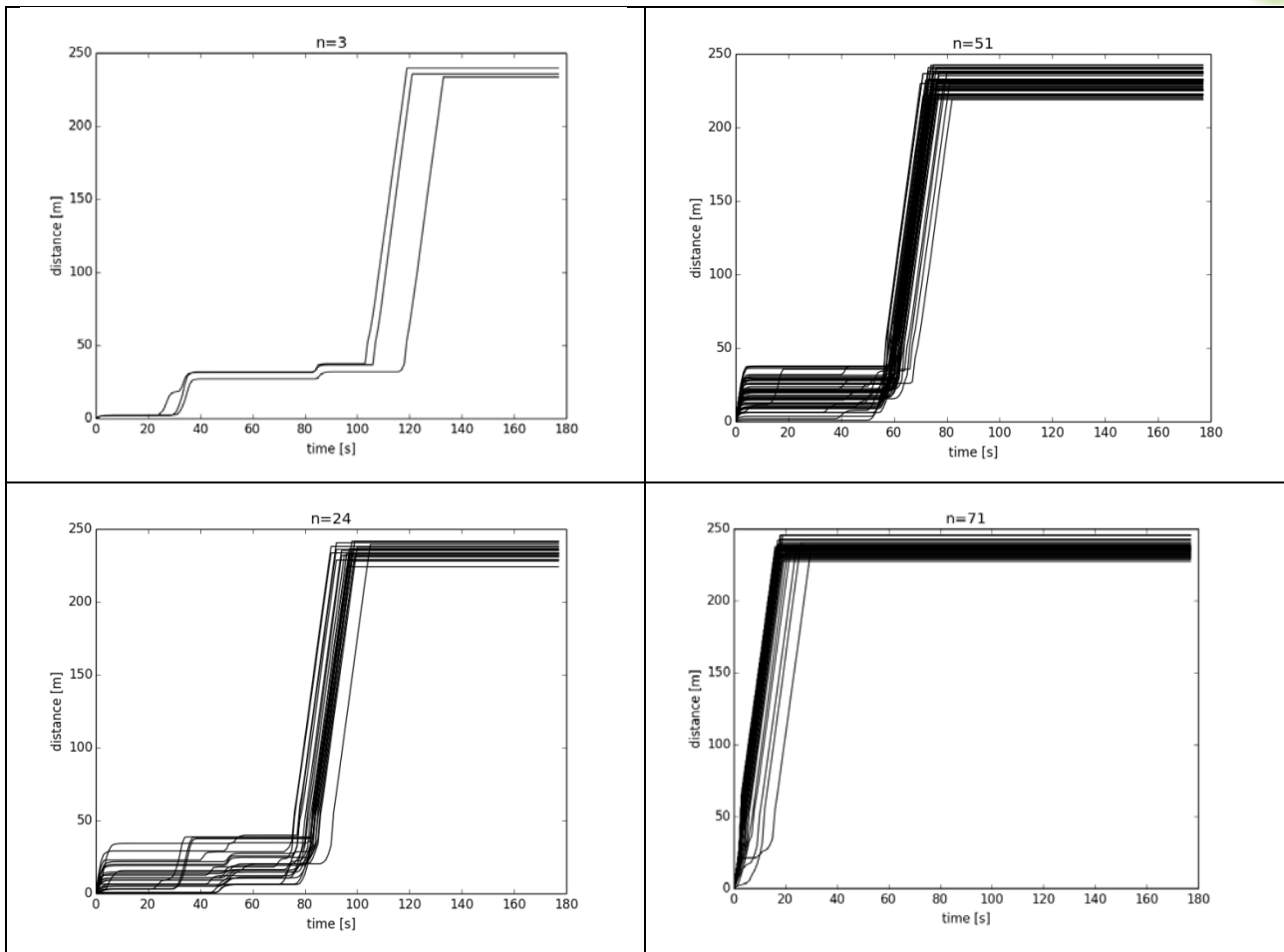


Figure 49: Results of the clustering approach with the clustering distance over time

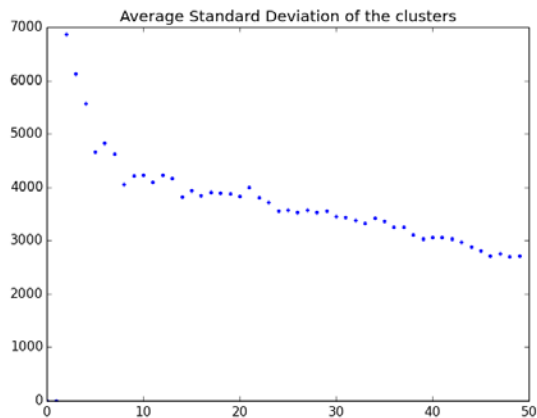


Figure 50: Average Standard Deviation of the clusters for the clustering over time

3.2.2.6 Summary of the Approaches

The evaluation of the clustering approaches with 2 to 49 clusters showed that the basic clustering approach had the best results for the average standard deviation of the emissions within the clusters. Therefore the simulation was performed only with the basic clustering approach and also with the Basic approach without clustering to compare the results.

3.2.3 Simulation Results

The ground truths of the “real” produced pollutant emission were collected to compare them with the estimated emissions (“real” is means the produced pollutant emissions for all vehicles in the simulation). For computing all emissions PHEMlight is used. For each simulation scenario the relative error of the estimated emissions was calculated as followed:

$$Error = \frac{(E_{approx.} - E_{real})}{E_{real}}$$

Where:

Error: Relative Error

E_{approx.}: Approximated emissions

E_{real}: real Emissions

3.2.3.1 Basic Approach

The simulation results show that even with a low penetration rate of 1 % the relative error of the estimated emissions is around 5 % after 1 hour of simulation and emission collecting time (see Figure 51). The relative error is decreasing with higher penetration rates, this effect can be expected because with a higher penetration rate the knowledge about the produced emissions is also larger.

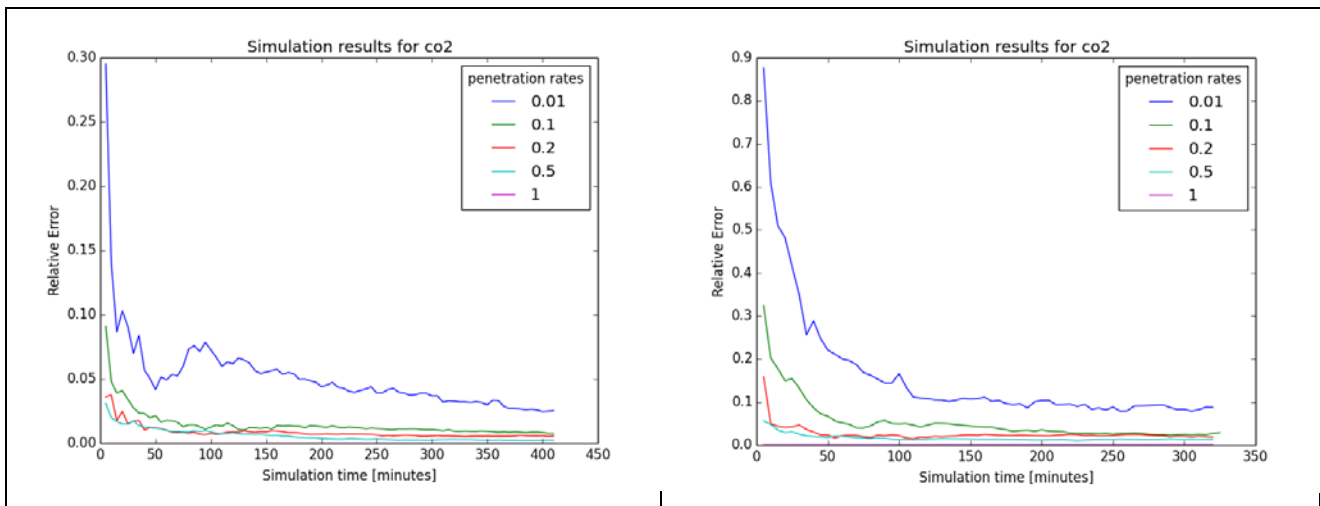


Figure 51: Relative Error of the estimated emissions over time (left). With different vehicle types (right)

3.2.3.2 Clustering Approach

For the emission monitoring the emission clusters are mapped onto the induction loops or street of the network. According to the assumption that the GPS positions of the CAMs are not accurate enough to know the exact lane and loop a vehicle is driving at, this mapping distinguishes only between the approaching streets, but not the lanes. The errors for the emission calculation with the clustering approach can be seen in Figure 52. The errors are slightly higher than the results for the basic approach but are still relatively good. When comparing the result for the simulation with different vehicle types the clustering approach performs much better than the basic approach (see Figure 52).

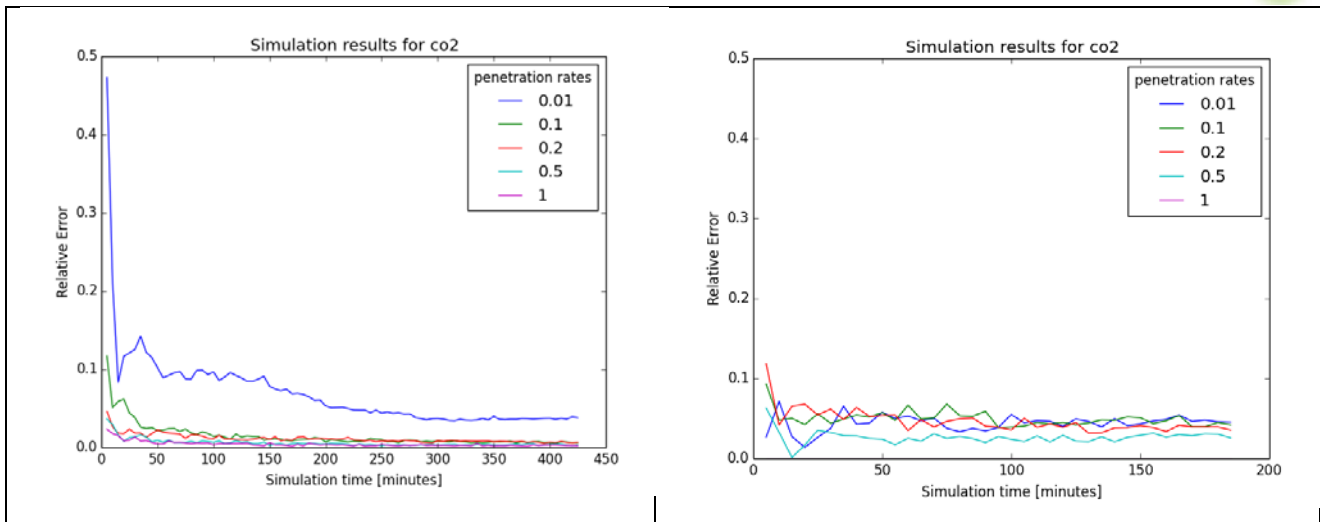


Figure 52: Relative Error of the estimated emissions over time for the clustering algorithm (left). With different vehicle types (right)

3.2.4 Discussions

In this work a combined approach to monitor emissions was presented and evaluated. The basic idea of the approach was to use the vehicle counts from induction loops and combine this information with the speed profile of equipped vehicles. The simulation results show that the algorithm can be used for emission monitoring. Even with low penetration rates of 1 % an error less than 10 % can be reached.

But the current state of research has some limitations. The simulation scenario which was used for the evaluation is an artificial one. To have a more realistic traffic situation the approach should also be simulated for an intersection with real world data (an existing intersection and its traffic demand). For this evaluation only simulation results have been considered. Simulation models are normally a simplification of a real world situation. Therefore it can be assumed that the simulated emissions will have a larger error when comparing to measured emissions in real world. Hence, the results of the algorithm should also be compared to real world measurements.

The described approaches was evaluated for monitoring a single intersection, but further research has to be done if the approach should be extended for larger regions. One solution could be that CAMs are collected inside the vehicle over a longer time and all the information is send via GSM or when a road side unit is available. In this case it is not needed that every intersection is equipped with a road side unit for emission monitoring, but the additional cost for GSM has to be considered.

4 Conclusion

For tactical purposes, such as adaptation of traffic lights to changes in traffic demand, traffic management needs on-line information about the state of the managed roads. Nowadays, such applications are realized using stationary detectors that are connected to a traffic light.

Cooperative traffic surveillance systems available on the market aim rather at offering traffic surveillance to a larger population of individuals. This development ignores the needs of traffic management, although traffic management is the primary actor to mitigate traffic congestions. COLOMBO's traffic surveillance system aims at bringing another evolution into the market, offering solutions from traffic management and road infrastructure providers.

Therefore, the employment of traffic surveillance systems based on vehicular communication was under evaluation within COLOMBO. The major motivation is to offer traffic management cheap and extendible solutions Decentralized Floating Car Data (DFCD). The unique approach of COLOMBO is to consider early penetration of vehicular communication (V2X), which will not include sufficient V2X-equipped vehicle for DFCD to provide accurate traffic state estimates. In [COLOMBO D1.2, 2014], we described four different DFCD solutions for collecting information about the state of monitored roads. We first considered a favorable high penetration of V2X-equipped vehicles (a.k.a class C vehicles), and then emphasized the impact of a reduced V2X penetration rate on the DFCD precision.

In this document, we propose to compensate missing V2X traffic sensing devices by integrating traffic states obtained from alternate technologies, such as WiFi-Direct or Bluetooth, into robust fusion engines

The first presented solution suggests to use smartphones equipping a large amount of drivers on the Cluster-based traffic surveillance approach described in [COLOMBO D1.2, 2014]. Smartphones have efficient communication capabilities, but a reduced GNSS precision. The presented solutions illustrated how to rely on map matching in order to compensate for lower precision GPS, and then integrate the mapped GNSS data into the Cluster-based traffic surveillance. Results showed to be able to closely monitor urban traffic at low V2X penetration through the use of smartphone.

The second presented solution proposes to use Bluetooth technology to provide additional sensing and dissemination capabilities. Considering the shorter Bluetooth transmit range, only trends in traffic density could be estimated and not absolute values. However, considering traffic speed, performance results showed to be able to closely estimate dynamic traffic speed at low V2X penetration and reasonable Bluetooth penetration rates.

The third presented DFCD solution suggested to use a Bayesian Network (BN) fusion approach to detect traffic at low V2X penetration, also considering a reasonable Bluetooth penetration rate of 30%. BNs are powerful tools enabling the handling and combination of incomplete data from various heterogeneous sources. Integrating as low as 1% V2X data into the BN fusion filter with error-prone vehicle detections by Bluetooth devices, managed to achieve accurate and reliable speed data for traffic and transportation management purposes.

Summarizing, this first part of the document described resilient methods for distributed traffic surveillance systems (DFCD) to cope with low penetration of V2X-equipped vehicles by integrating alternate data sources from alternate technologies. With this, COLOMBO provided modern and flexible methods for traffic state monitoring already at an early deployment of C-ITS systems. Another part of this document has been dedicated to build on the previously proposed resilient DFCD to detect traffic anomalies and incidents as well as locally modelling emissions.

An incident detection system has been designed, implemented and tested within COLOMBO. The system relies on trajectories that can be monitored by COLOMBO's DFCD. Within the scenario, it did not produce any false alarm and reliably detected every incident for a simulated V2X

penetration rate of 100%. The system exhibited notable robustness against noise added for simulating the position distortion of the GPS signal, as well as against reduced V2X penetration.

Finally, a combined approach to monitor emissions was presented and evaluated. The basic idea of the approach was to use the vehicle counts from induction loops and combine this information with the speed profile monitored by COLOMBO's resilient DFCD. The simulation results show that the algorithm can be used for emission monitoring. Even with low penetration rates of 1 % an error less than 10 % can be reached.

Appendix A – References

- [1] Jan Fuglestad, Terje Berntsen, Gunnar Myhre, Kristin Rypdal, and Ragnhild Bieltvedt Skeie. Climate forcing from the transport sectors. Proceedings of the National Academy of Sciences, 105(2):454–458, 2008.
- [2] Ulrich Radl. Bewertungsmethodik der Verkehrssteuerung innerstädtischer Hauptverkehrsstraßen. Straßenverkehrstechnik, 59:484–486, July 2015.
- [3] M. Carriero A. Krasenbrink L. Rubino, P. Bonnel. Portable emission measurement system (pems) for heavy duty diesel vehicle pm measurement: The european pm pems program. SAE International Journal of Engines, 118(3):17, 2009. <http://papers.sae.org/2009-24-0149/>.
- [4] A. Bigazzi, J.W.C. Van Lint, G. Klunder, U. Stelwagen, and N. Ligterink. Traffic data for local emissions monitoring at a signalized intersection. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 210–215, Sept 2010.
- [5] Astrid Gühnemann, Ralf-Peter Schäfer, Kai-Uwe Thiessenhusen, and Peter Wagner. Monitoring traffic and emissions by floating car data. Technical report, 2004.
- [6] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo—simulation of urban mobility. International Journal On Advances in Systems and Measurements, 5(3&4), 2012.
- [7] Daniel Krajzewicz, Marek Heinrich, Michela Milano, Paolo Bellavista, Thomas Stützle, Jérôme Härrä, Thrasyvoulos Spyropoulos, Robbin Blokpoel, Stefan Hausberger, and Martin Fellendorf. Colombo: investigating the potential of v2x for traffic management purposes assuming low penetration rates. ITS Europe, 2013.
- [8] Luz R. and Hausberger S. User guide for the model phem, version 11.2, 2013.
- [9] Intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service, 2011.
- [10] Daniel Müllner. fastcluster: Fast hierarchical clustering routines for R and Python.
- [11] Daniel Müllner. Fast hierarchical, agglomerative clustering routines for r and python. Journal of Statistical Software, 53(9):1pp, 2013. <http://www.jstatsoft.org/v53/i09/>.
- [12] Meysel, F., Map-based Multi-Object Dynamic Scene Description and Atypical Event Detection. Ph.D. thesis, Humboldt Universität zu Berlin, 2008.
- [13] Leich, Andreas and Junghans, Marek and Kozempel, Karsten and Saul, Hagen (2015) Road user tracker based on robust regression with GNC and preconditioning. In: Video Surveillance and Transportation Imaging Applications 2015, 9407. SPIE. IS&T/SPIE Electronic Imaging 2015, 8.2.2015 - 12.2.2015, San Francisco. DOI: 10.1117/12.2082520. ISBN 9781628414974. ISSN 0277-786X
- [14] Saul, Hagen and Kozempel, Karsten and Haberjahn, Mathias (2014) A Comparison of Methods for Detecting atypical Trajectories. In: Urban Transport XX, 138, Seiten 393-403. WITPress. 20th International Conference on Urban Transport and the Environment, 28.-30. Mai 2014, Portugal. ISBN 978-1-84564-778-0. ISSN 1746-4498
- [15] Yacoob, Y. and Black, M.J., Parameterized modeling and recognition of activities. Proceedings of the Sixth International Conference on Computer Vision, IEEE Computer Society, ICCV '98, pp. 120-, 1998.

- [16] Bashir, F.I., Khokhar, A.A. and Schonfeld, D., Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Transactions on Multimedia*, pp. 58-65, 2007.
- [17] Bashir, F.I., Khokhar, A.A. & Schonfeld, D., Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. *Image Processing, IEEE Transactions on*, 16, pp. 1912-1919, 2007.
- [18] Schölkopf, Bernhard, and Alexander J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [19] Picciarelli, C., Micheloni, C. & Foresti, G.L., Trajectory-based anomalous event detection. *IEEE Trans Circuits Syst Video Techn*, 18, pp.1544-1554, 2008.
- [20] Chen, Y., Liang, G., Lee, K. K., & Xu, Y. (2007, July). Abnormal behavior detection by multi-svm-based bayesian network. In *Information Acquisition, 2007. ICIA'07. International Conference on* (pp. 298-303). IEEE.
- [21] Oliver, N., Rosario, B. and Pentland, A., A bayesian computer vision system for modeling human interactions. *IEEE Trans Pattern Anal Mach Intell*, 22, pp. 831-843, 2000.
- [22] Natarajan, P. and Nevatia, R., Coupled hidden semi markov models for activity recognition. *Proceedings of the IEEE Workshop on Motion and Video Computing, IEEE Computer Society: Washington, DC, USA, WMVC '07*, pp. 10-, 2007.
- [23] Johnson, N. and Hogg, D., Learning the distribution of object trajectories for event recognition. *Image Vision Comput*, 14, pp. 609-615, 1996.
- [24] Sumpter, N. and Bulpitt, A.J., Learning spatio-temporal patterns for predicting object behaviour. *Image Vision Comput*, 18, pp. 697-704, 2000.
- [25] Owens, J. and Hunter, A., Application of the self-organizing map to trajectory classification. *Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, IEEE Computer Society: Washington, DC, USA, pp. 77-, 2000.
- [26] Vaswani, N., Chowdhury, A.R. and Chellappa, R., Shape activity: A continuous state hmm for moving/deforming shapes with application to abnormal activity detection, 2005.
- [27] Hervieu, A., Bouthemy, P. and Cadre, J.P.L., A statistical video content recognition method using invariant features on object trajectories. *IEEE Trans Circuits Syst Video Techn*, 18, pp. 1533-1543, 2008.
- [28] Berclaz, J., Fleuret, F. and Fua, P., Multi-camera tracking and atypical motion detection with behavioral maps. *ECCV* (3), pp. 112-125, 2008.
- [29] Nascimento, J.C., Figueiredo, M.A.T. and Marques, J.S., Trajectory classification using switched dynamical hidden markov models. *IEEE Transactions on Image Processing*, 19, pp. 1338-1348, 2010.
- [30] Benezeth, Y., Jodoin, P.M. and Saligrama, V., Abnormality detection using low-level co-occurring events. *Pattern Recogn Lett*, 32, pp. 423-431, 2011.
- [31] Hongeng, S. and Nevatia, R., Multi-agent event recognition. *ICCV*, pp. 84-93, 2001.
- [32] Cai, Y., Wang, H., Chen, X., and Jiang, H., Trajectory-based anomalous behaviour detection for intelligent traffic surveillance. *IET Intell. Transp. Syst.*, 2015, Vol. 9, Iss. 8, pp. 810-816, 2015
- [33] Zou, X., and Levinson, D. (2006). Modeling Pipeline Driving Behaviors: Hidden Markov Model Approach. *Transportation Research Record: Journal of the Transportation Research Board*, (1980), 16-23.

- [34] Morris, B. T., & Trivedi, M. M. (2011). Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11), 2287-2301.
- [35] Fawcett, T. An introduction to ROC analysis. In *Pattern recognition letters* 27.8 (2006), pp. 861– 874.
- [36] Bauer, S., Obst, M., Wanielik, G. 3D environment modeling for GPS multipath detection in urban areas. In *Systems, Signals and Devices (SSD)*, 2012 9th International Multi-Conference on , vol., no., pp.1-5, 20-23 March 2012, doi: 10.1109/SSD.2012.6198054
- [37] Rankin, J. GPS and differential GPS: an error model for sensor simulation. *Position Location and Navigation Symposium. IEEE*, 1994.
- [38] Clark, J. J.; Yuille, A. L.: *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers Boston/Dordrecht/London, 1990.
- [39] R. Hoyer; C. Leitzke; M. Schäfer: Reisezeitbestimmung mittels Bluetooth-Technologie, in *Straßenverkehrstechnik*, Heft 07/2011, S. 417 – 426; Kirschbaum Verlag, Bonn, 2011.
- [40] Mäkinen, T. et al.: DRIVE C2X methodology framework. Deliverable D22.1, 29.09.2011.
- [41] Neapolitan, R.: *Learning Bayesian Networks*. Pearson Education, Inc., Prentice Hall, 2004.
- [42] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Second Printing, Morgan Kaufmann Publishers, Inc., San Francisco (CA), 1991.
- [43] Strang, T.: *European C2C Activities*, 2010.
- [44] Holzmann, C. and Oppl, S.: *Bluetooth in a Nutshell*, JKU Linz, 2003.
- [45] Pál, I. and Sauerländer-Biebl, A., *Automatische multimodale Verkehrsmoduserkennung und Situationserfassung mit Hilfe von Fuzzy-Regeln*. 2014
- [46] EU Drive C2X: <http://www.drive-c2x.eu/publications>
- [47] RiLSA1 2014: http://sourceforge.net/projects/sumo/files/traffic_data/scenarios/RiLSA/#
- [48] Junghans, M. and Bauer, S. and Kozempel, K. and Schulz, J. and Haberjahn, M. and Lubert, A. and Schmietendorf, G. and Wagner, P.: *Alternative Erfassungsmethoden sind analysiert und getestet – Meilensteinbericht des Projektes Verkehrs- und Mobilitätsmanagement im AP 2100 Bildverarbeitung*. Version 1.1, January 23, 2011.
- [49] Weinzerl, J.; Fast, A.: *Verkehrsablauf erkennen durch intelligente Sensoren auf Autobahnen und in urbanen Gebieten*, Beitrag zur Heureka 2011, Graz, 2011.
- [50] A. Bhaskar; E. Chung: Fundamental Understanding on the Use of Bluetooth Scanner as a Complementary Transport Data, in *Transportation Research Part C* 37 (2013), S. 42 – 72, Elsevier Ltd., 2013.
- [51] T. Otterstätter: *Methoden zur Erfassung von Verkehrsströmen und Fahrzeiten mit stationären Detektoren Dissertation*, Universität Stuttgart, 2013.
- [52] Y. Bar-Shalom and X.-R. Li and T. Kirubarajan: *ESTIMATION WITH APPLICATIONS TO TRACKING AND NAVIGATION ALGORITHMS AND SOFTWARE FOR INFORMATION EXTRACTION*. ISBN 0-9648312-1-X, 2001.
- [53] Roure, J.; Sanguesa, R.: *Incremental Methods for Bayesian Network Learning*. In *Uncertainty inCohen, I.; Bronstein, A.; Cozman, F. G.: Adaptive Online Learning of Bayesian Network*
- [54] Cohen, I.; Bronstein, A.; Cozman, F. G.: *Adaptive Online Learning of Bayesian Network Parameters*, <http://www.hpl.hp.com/techreports/2001/HPL-2001-156.html>, 2001.

[55] EU iTetris: <http://www.ict-itetris.eu/simulator/>

[56] EU COLOMBO: <http://www.colombo-fp7.eu/>

Appendix B: Parameterization of Kalman Filter

For post processing of the trajectories with noise added, a Kalman filter with the following parameters was used:

The state vector including position und velocity. Therefore the following state vector, state transition matrix and state Covariance matrices were used:

$$X = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} ; \quad F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad S_t = \begin{bmatrix} \sigma_{\dot{x}_t}^2 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{y}_t}^2 & 0 & 0 \\ 0 & 0 & \sigma_{x_t}^2 & 0 \\ 0 & 0 & 0 & \sigma_{y_t}^2 \end{bmatrix}$$

It is assumed, that only the vector position can be measured. Therefore the measurement vector and observation matrix were defined as follows:

$$Y_t = \begin{bmatrix} \tilde{x}_t \\ \tilde{y}_t \end{bmatrix} ; \quad M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The Covariance matrix and the influence matrix of the process noise were set as:

$$Q = \begin{bmatrix} \sigma_{n_1}^2 & 0 \\ 0 & \sigma_{n_2}^2 \end{bmatrix} ; \quad W = \begin{bmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

and the Covariance matrix for measurement noise as:

$$R = \begin{bmatrix} \sigma_{n_1}^2 & 0 \\ 0 & \sigma_{n_2}^2 \end{bmatrix}$$

The formula for state transition (also prediction and a priori state) is straightforward:

$$X_{t,t-1} = F X_{t-1,t-1}$$

The a priori covariance is computed via:

$$S_{t,t-1} = F S_{t-1,t-1} F^T + W_t Q W_t^T$$

Then the Kalman Gain is:

$$K_t = S_{t,t-1} M^T [M S_{t,t-1} M^T + R]^{-1}$$

And the correction (or a posteriori) state is:

$$X_{t,t} = X_{t,t-1} + K_t (Y_t - M X_{t,t-1})$$

With the a posteriori covariance:

$$S_{t,t} = [I - K_t M] S_{t,t-1}$$

The initial state covariances were set to:

$$\begin{aligned}\sigma_{x_t} &= 0.01 \text{ m} \\ \sigma_{y_t} &= 0.01 \text{ m} \\ \sigma_{\dot{x}_t} &= 1 \text{ m/s} \\ \sigma_{\dot{y}_t} &= 1 \text{ m/s} \\ S_t &= \begin{bmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

Measurement noise covariances were set to:

$$\begin{aligned}\sigma_{n_1} &= 3 \text{ m} \\ \sigma_{n_2} &= 3 \text{ m} \\ R &= \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}\end{aligned}$$

And the process noise covariances were set to:

$$\begin{aligned}\sigma_{n_1} &= 0.1 \text{ m} \\ \sigma_{n_2} &= 0.1 \text{ m} \\ Q &= \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}\end{aligned}$$